
Eine Referenzarchitektur für die assistierte und automatisierte Fahrzeugführung mit Fahrereinbindung

Vom Fachbereich Maschinenbau an der
Technischen Universität Darmstadt
zur Erlangung des Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte

Dissertation

vorgelegt von

Dipl.-Ing. Felix Gerhard Otto Lotz
aus Alzenau

Berichterstatter: Prof. Dr. rer. nat. Hermann Winner

Mitberichterstatter: Prof. Dr.-Ing. Jürgen Adamy

Tag der Einreichung: 20.10.2016

Tag der mündlichen Prüfung: 11.01.2017

Darmstadt 2017

D 17

Vorwort

Diese Arbeit entstand während und im Anschluss meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Fachgebiet Fahrzeugtechnik der Technischen Universität Darmstadt. Die Inhalte der Dissertation wurden im Rahmen der Forschungs Kooperation PRORETA 3 in Zusammenarbeit mit der Continental AG erarbeitet.

Besonderer Dank gilt meinem Doktorvater Herrn Prof. Dr. rer. nat. Hermann Winner, der durch viele inhaltliche Diskussionen und durch das Schaffen einer durch gegenseitiges Vertrauen gekennzeichneten Arbeitsatmosphäre wesentlich zum Gelingen dieser Arbeit beigetragen hat.

Herrn Prof. Dr.-Ing. Jürgen Adamy danke ich für die bereitwillige Übernahme des Korreferats sowie für das motivierende und konstruktive Feedback im Rahmen der PRORETA 3 Projekttrunden und darüber hinaus.

Meinen Teamkameraden Eric Bauer, Dr. Matthias Pfromm und Dr. Matthias Schreier danke ich nicht nur für den ständigen fachlichen Austausch und die sehr gute Zusammenarbeit, sondern auch - und insbesondere - für die großartige und durch Freundschaft geprägte gemeinsame Zeit. Dies gilt in gleicher Weise für das gesamte Team des Fachgebietes Fahrzeugtechnik.

Für das Initiieren und Betreuen des Projektes seitens der Continental AG möchte ich stellvertretend Herrn Alfred Eckert, Herrn Dr. Stefan Lüke und Herrn Stephan Cieler ganz herzlichen Dank ausrichten. Darüber hinaus bin ich allen Mitarbeitern der Continental Zukunftsentwicklung in Frankfurt für die hervorragende Unterstützung in allen Belangen zu großem Dank verpflichtet und freue mich, dass ich im Anschluss an meine Projektzeit Teil dieses Teams werden durfte.

Allen beteiligten Studenten danke ich für ihren Einsatz und ihr Engagement und den damit einhergehenden wichtigen Beitrag zu dieser Arbeit.

Nicht möglich gewesen wäre diese Dissertation ohne meine Familie und insbesondere meine Eltern, die mich all die Jahre durch Studium und Promotionszeit hindurch bedingungslos unterstützt und gefördert haben.

Meiner Frau Marisa danke ich nicht zuletzt für ihre liebevolle Unterstützung. Erst durch ihr Verständnis und ihren Rückhalt war es möglich, der Doppelbelastung nach meiner Projektzeit standzuhalten.

Felix Lotz

Darmstadt im Oktober 2016

Inhaltsverzeichnis

Vorwort.....	III
Inhaltsverzeichnis	IV
Abkürzungen.....	VI
Formelzeichen und Indizes	VII
Kurzzusammenfassung	VIII
1 Einleitung.....	1
1.1 Motivation	1
1.2 Grundlagen zur Systemarchitektur	3
1.2.1 Begriffsdefinitionen	3
1.2.2 Zweck der Systemarchitektur	6
1.2.3 Architektur-Entwurfsprozess	7
1.2.4 Bewertungsverfahren von Systemarchitekturen	12
1.3 Entscheidungsverfahren komplexer Probleme	14
1.4 Ziele der Arbeit.....	16
1.5 Methodik und Aufbau der Arbeit.....	19
2 Anforderungsanalyse.....	22
2.1 Das Projekt PRORETA 3	22
2.2 Funktionale Anforderungen.....	24
2.2.1 Anforderungsdefinition in der Fahrerassistenzsystem-Entwicklung	24
2.2.2 Szenarienkatalog	26
2.2.3 Zusammenfassung.....	31
2.3 Nichtfunktionale Anforderungen.....	35
2.3.1 Allgemeine nichtfunktionale Anforderungen	35
2.3.2 Nichtfunktionale Anforderungen für A/A-Systeme	36
2.3.3 Zusammenfassung.....	41
3 Funktionale Architekturen für automatisierte Fahrzeuge	45
3.1 Sequenzielle Basis-Architekturen	45
3.2 Parallele Basis-Architekturen.....	47
3.3 Hybride Basis-Architekturen.....	51
3.4 Anwendungsbeispiele für automatisierte Fahrzeuge.....	55
3.4.1 Ansätze nach Dickmanns und Maurer	56
3.4.2 Ansätze im Rahmen der DARPA Urban Challenge.....	57
3.4.3 Stadtpilot	58

3.4.4 Conduct-by-Wire.....	59
3.5 Zusammenfassung	60
4 Auswahl der Basis-Architektur.....	61
4.1 Varianten der Basis-Architektur	61
4.1.1 Sequenzielle Variante	61
4.1.2 Parallele Variante.....	64
4.1.3 Hybride Variante	67
4.2 Bewertung der Varianten und Auswahl der Basis-Architektur.....	69
5 Entwicklung der Referenzarchitektur	74
5.1 Festlegung der Systemhierarchie.....	74
5.1.1 Hierarchische Abstraktionsmodelle im Stand der Technik	74
5.1.2 Einflussfaktoren auf die Systemebenen	77
5.1.3 Identifikation der Systemebenen.....	80
5.2 Kopplung der Ebenen	85
5.2.1 Detaillierung der Kontrollhierarchie	85
5.2.2 Kontrollhierarchie zwischen Navigations- und Führungsebene	90
5.2.3 Kontrollhierarchie zwischen Führungs- und Stabilisierungsebene.....	100
5.2.4 Kontrollhierarchie innerhalb der Führungsebene.....	102
5.3 Modus-Entscheidung und -Überwachung	118
5.4 Weltmodell.....	122
6 Synthese.....	128
6.1 Bausteinsicht.....	128
6.2 Diskussion	134
6.3 Umsetzung im Projekt PRORETA 3	140
7 Fazit und Ausblick	149
A Anforderungen	151
A.1 Szenarienkatalog für Autobahnen.....	151
A.2 Szenarienkatalog für Landstraßen	154
A.3 Szenarienkatalog für Stadtstraßen	158
A.4 Szenarienkatalog für assistiertes und unfallvermeidendes Fahren	161
B Schnittstellenbeschreibung.....	162
Literaturverzeichnis	166
Eigene Veröffentlichungen	176
Betreute studentische Arbeiten	177
Lebenslauf.....	178

Abkürzungen

Abkürzung	Beschreibung
<i>A/A-System</i>	Assistenz- und Automatisierungssystem
<i>ACC</i>	Adaptive Cruise Control
<i>ASIL</i>	Automotive Safety Integrity Level
<i>ATAM</i>	Architecture Tradeoff Analysis Method
<i>CbW</i>	Conduct-by-Wire
<i>CD</i>	Central Decision
<i>DAMN</i>	Distributed Architecture for Mobile Navigation
<i>DARPA</i>	Defense Advanced Research Projects Agency
<i>E/E</i>	Elektrik / Elektronik
<i>EMS-Vision</i>	Expected-based Multi-focal Saccadic Vision
<i>EPS</i>	EPS: Electric Power Steering
<i>ESC</i>	Electronic Stability Control
<i>FE</i>	Führungsebene
<i>FPK</i>	Frei programmierbares Kombiinstrument
<i>GDF</i>	Geographic Data Files
<i>LKA</i>	Lane Keeping Assistance
<i>MMS</i>	Mensch-Maschine-Schnittstelle
<i>NASREM</i>	NASA Standard Reference Model (Architecture)
<i>NDS</i>	Navigation Data Standard Format
<i>NE</i>	Navigationsebene
<i>NMPC</i>	Non-linear model-predictive control
<i>Prometheus</i>	Programme for a European Traffic of Highest Efficiency and Unprecedented Safety
<i>QAS</i>	Qualitätsattributsszenario
<i>RAA</i>	Richtlinie für die Anlage von Autobahnen
<i>RAL</i>	Richtlinie für die Anlage von Landstraßen
<i>RAS_t</i>	Richtlinie für die Anlage von Stadtstraßen
<i>RCS</i>	Real-Time Control System
<i>RNDF</i>	Route Network Definition File
<i>SAAM</i>	Software Architecture Analysis Method
<i>SE</i>	Stabilisierungsebene
<i>SM_{PA}</i>	Sense-Model-Plan-Act
<i>SP_A</i>	Sense-Plan-Act
<i>StVO</i>	Straßenverkehrsordnung
<i>TTC</i>	Time-To-Collision
<i>UI</i>	User Interface (Nutzerschnittstelle)
<i>UML</i>	Unified Modeling Language
<i>V2X</i>	Vehicle-To-X
<i>VaMoRs</i>	Versuchsfahrzeug für autonome Mobilität und Rechnersehen

Formelzeichen und Indizes

Symbol	Einheit	Beschreibung
<i>M</i>	-	Menge
<i>FS_befahren</i>	-	Manöver-Element: Fahrstreifen befahren
<i>FSW</i>	-	Manöver-Element: Fahrstreifenwechsel

Index	Beschreibung
<i>Auswahl</i>	Auswahl (ausgewähltes Manöver)
<i>durchführbar</i>	Durchführbar (durchführbares Manöver)
<i>links</i>	links
<i>rechts</i>	rechts
<i>Zielerreichung</i>	Zielerreichung (zur Zielerreichung notwendiges Manöver)

Kurzzusammenfassung

Basierend auf den Erfolgen der Fahrerassistenzsystem-Entwicklung der vergangenen Jahre und dem vielversprechenden Ausblick auf zukünftige Geschäftsfelder und Nutzungsmöglichkeiten rund um das Automobil besitzt das Thema des automatisierten Fahrens weltweit einen hohen Stellenwert bei Fahrzeug-Herstellern, Zulieferern sowie IT-Unternehmen. Eine der größten Herausforderungen dabei ist die Komplexitätsbeherrschung sowie die Ermöglichung eines Sicherheitsnachweises solcher Systeme, wobei die Systemarchitektur dafür maßgeblich ist.

Gegenstand der vorliegenden Arbeit ist die Entwicklung einer funktionalen Systemarchitektur, die den Anforderungen des assistierten, teilautomatisierten bis hin zum vollautomatisierten Fahrens gerecht werden soll. Dabei steht insbesondere die Architektur als wissenschaftliche Disziplin im Vordergrund, in der Entscheidungsalternativen erarbeitet und die Entscheidungen durch Abwägung der sich daraus ergebenden Konsequenzen bewertet und dokumentiert werden. Um dieses Ziel zu erreichen wird die Architekturentwicklung im Rahmen dieser Arbeit als ein komplexes Entscheidungsproblem aufgefasst, das durch Kombination von Methoden heuristischer Entscheidungsverfahren und bekannter Architekturentwicklungsverfahren schrittweise gelöst wird.

Im ersten Schritt erfolgt eine Anforderungsanalyse, in der die funktionalen Systemanforderungen in Form von notwendigen Verhaltensweisen (Fahrmanöver) systematisch hergeleitet werden. Des Weiteren werden relevante nichtfunktionale Anforderungen an die Architektur identifiziert, wobei insbesondere auf die Testbarkeit sowie Änder- bzw. Erweiterbarkeit eingegangen wird und somit die Grundlage für die Bewertungskriterien darstellen.

Im zweiten Schritt erfolgt eine Recherche geeigneter Basis-Architekturvarianten aus der Robotik, die die grundlegende Systemstruktur beschreiben. Im Rahmen der darauf folgenden Bewertung der Varianten wird die hybride Basisarchitektur als am geeignetsten für das zugrunde liegende Problem ausgewählt, da im Gegensatz dazu die sequenzielle Variante aufgrund der Nichterfüllbarkeit der funktionalen Anforderungen ausscheidet und die parallele Variante durch die notwendigen komplexen Kontroll-Arbitrierungsmechanismen und geringerer Vorhersagbarkeit des resultierenden Verhaltens einen Zielkonflikt zwischen Test- und Erweiterbarkeit erzeugt.

Im vierten Schritt erfolgt schließlich die Entwicklung der Referenzarchitektur, beginnend mit einer Festlegung des zugrunde liegenden hierarchischen Abstraktionsmodells. Das Modell der Fahrzeugführung nach Donges wird dabei als am geeignetsten zur

Strukturierung der funktionalen Architektur automatisierter Fahrzeuge angesehen, da es die hierarchische Struktur der Umgebung des Systems (Topologie des Straßenverkehrs) besser auf mögliche funktionale Zuständigkeiten der Systemebenen abbilden kann.

Auf dieser Basis erfolgt eine Detaillierung der Kontrollhierarchie des Gesamtsystems beginnend von der Routenplanung bis hin zur Fahrdynamikregelung unter Betrachtung dafür notwendiger Informationsverarbeitungsschritte und deren jeweiliger Entscheidungsspielräume. Durch Variation der Zuordnung dieser zu den drei Hierarchieelementen Navigationsebene, Führungsebene und Stabilisierungsebene können systematisch verschiedene Architekturvarianten generiert und schließlich bewertet werden. Von besonderer Bedeutung dabei ist das Zusammenspiel zwischen deliberativen, planbaren Systemelementen zur Zielerreichung einerseits und reaktiven Systemelementen zur schnellen Reaktion auf sich ändernde Situationsparameter andererseits. Im Rahmen der Variantenerstellung zur Auftrennung der dafür wichtigen Führungsebene fällt dabei auf, dass sich insbesondere im Kontext des stark strukturierten und reglementierten Umfelds automatisierter Fahrzeugführungssysteme die reaktiven und deliberativen Systemanteile nicht eindeutig voneinander trennen lassen. Dies wird insbesondere bei kooperativen / taktischen Fahrmanövern deutlich, bei denen die Verhaltensentscheidung stark abhängig von der Verhaltensausführung auf Trajektorien-Ebene ist, um eine Kollisionsfreiheit der Verkehrsteilnehmer sicherzustellen.

Aus diesem Grund - und um eine über alle ermittelten Szenarien konsistente Architekturbeschreibung zu ermöglichen - wird im Rahmen der vorliegenden Arbeit eine zweigeteilte Führungsebene empfohlen. In der oberen Schicht erfolgt eine Verhaltens- bzw. Fahrmanöverplanung, die bereits die Kollisionsfreiheit und somit einen Teil der Reaktivität des Gesamtsystems sicherstellt. Als Schnittstelle zur unterlagerten Trajektorienplanung muss sie somit eine explizite Angabe der zulässigen räumlichen und zeitlichen Entscheidungsspielräume zur Verfügung stellen. Zusätzlich besteht bei dieser Architekturvariante die Möglichkeit der Schaffung einer strukturellen Redundanz, sofern der Trajektorienplanungs-Algorithmus ebenfalls reaktiv die Kollisionsfreiheit des Fahrzeugs sicherstellt. Neben der Festlegung der Kontrollhierarchie wird zusätzlich der Informationsbedarf der Planungsmodule in Richtung des Umfeldmodells skizziert sowie die notwendigen Mensch-Maschine-Schnittstellen zur Fahrereinbindung.

Die ermittelte Referenzarchitektur dient jedoch nicht nur als Vorlage für ähnliche Problemstellungen. Durch Offenlegung des Erkenntniswegs, der zur ermittelten Referenzarchitektur führt, bietet sie dem Leser darüber hinaus und im Gegensatz zum Großteil der in der Literatur bekannten Veröffentlichungen eine bessere Einschätzbarkeit der Übertragbarkeit der Lösung auf die eigene Problemstellung und stellt dabei die Architektur als Disziplin in den Vordergrund.

1 Einleitung

1.1 Motivation

Die Fahrzeugindustrie steht nach eigener Einschätzung aktuell vor einem der größten Umbrüche seit Bestehen des Automobils¹. Die zunehmende Digitalisierung und Vernetzung von Fahrzeugen und Fahrzeugfunktionen eröffnen eine Vielzahl neuer Möglichkeiten, das Fahren noch sicherer und komfortabler zu machen. Dass vor allem die Fahrzeugautomatisierung sowie Systeme der aktiven Sicherheit einen Löwenanteil am zukünftigen Marktpotenzial ausmachen werden, prognostiziert eine Studie der Unternehmensberatung Strategy&: So wird sich der Markt für digitale Produkte allein im Pkw-Segment zwischen 2015 und 2020 von ca. 32 Mrd. Euro auf geschätzte 115 Mrd. Euro mehr als verdreifachen, wovon alleine ca. 47 Mrd. Euro auf Sicherheitssysteme sowie ca. 36 Mrd. Euro auf die Fahrzeugautomatisierung entfallen².

Diese Entwicklung stützt sich vor allem auf die technischen Fortschritte, die in der Forschung und Entwicklung von (umfelderfassenden) Fahrerassistenzsystemen in den vergangenen Jahren erreicht werden konnten und die bereits heute einen wichtigen Anteil in der Wertschöpfungskette der Automobilindustrie darstellen³. Auch neue Anforderungen aus Verbraucherschutz und Gesetzgebung, wie z. B. die Berücksichtigung von aktiv eingreifenden Fahrerassistenzsystemen in der Bewertung des Euro NCAP⁴ oder die verpflichtende Ausrüstung von Notbremssystemen bei Lkw seit 2013⁵, tragen zur Weiterentwicklung von Assistenzsystemen und deren Verbreitung bis in das Kleinwagensegment bei.

Insbesondere die jüngsten Entwicklungen im Bereich des automatisierten Fahrens zeigen, dass auch neue „Spieler“ das Ziel haben, diesen Markt zu erschließen und mitzugestalten, wie sich z. B. an der umfassenden medialen Berichterstattung des „Google self-driving car“ ablesen lässt⁶. Die etablierten Automobilhersteller und -Zulieferer

¹ Hawranek & Schulz (2014): Die zweite Erfindung des Automobils, S. 16

² Viereckl et al. (2014): Racing Ahead - The Connected C@r 2014 Study

³ Winner (2015): Quo vadis, FAS?, S. 1168

⁴ Euro NCAP (2016): <http://de.euroncap.com>, letzter Zugriff am 02.10.16

⁵ EU (2009): Verordnung (EG) Nr. 661 / 2009 des Europäischen Parlamentes und des Rates

⁶ siehe z. B. Hawranek & Schulz (2014): Die zweite Erfindung des Automobils, S. 14-21

stehen somit unter Zugzwang, diese Entwicklung nicht zu „verschlafen“, die das Potenzial mit sich bringt, innerhalb weniger Produktzyklen das Nutzungs- und Bedienverhalten so stark zu verändern, dass die Marktfähigkeit heutiger Fahrzeuge und somit deren Hersteller in Frage gestellt werden könnte^{7a 8}.

Dem skizzierten Spannungsfeld gehen vielschichtige entwicklungsbegleitende Herausforderungen einher, wobei jene technologischer Art nicht unbedingt als die gravierendsten angesehen werden: Um insbesondere haftungs- und zulassungsrechtliche Hürden überwinden zu können, muss ein Nachweis der Sicherheit automatisierter Fahrzeuge geführt werden. Hier sehen viele Experten den eigentlichen kritischen Pfad in der Entwicklung, denn eine Test- und Freigabemetrik für solche Systeme existiert derzeit noch nicht^{7b}.

Diese Erkenntnisse lassen u.a. zwei Tendenzen in der Entwicklung automatisierter Fahrzeugführungssysteme erkennen: Trotz eines im Vergleich zu heutigen Fahrerassistenzsystemen größeren Funktionsumfangs sollte zum einen die Gesamtsystemkomplexität automatisierter Fahrzeuge möglichst nicht im gleichen Umfang (oder sogar überproportional) wachsen, da dadurch auch der gesteigerte Entwicklungs- und vor allem der Test- und Freigabeaufwand das obige Problem weiter verschärfen würden. Zum anderen deutet sich an, dass eine Verschiebung der Aufgabe des Menschen vom aktiven Fahrer hin zum Passagier nicht innerhalb weniger Produktzyklen, sondern vielmehr längerfristig und schrittweise erfolgen könnte.

In diesem Zusammenhang kommt der funktionalen Systemarchitektur automatisierter Fahrzeuge eine zentrale Rolle zu: Sie bestimmt maßgeblich die Gesamtsystemkomplexität und stellt somit den Haupteinflussfaktor für eine effektive und effiziente Systementwicklung dar. Ziel der vorliegenden Arbeit ist es, eine begründete Aussage zu treffen, wie eine solche funktionale Systemarchitektur beschaffen sein sollte, die ein funktionales Spektrum vom assistierten bis hin zum vollautomatisierten Fahren beherrscht und somit die Option, den Fahrer als wichtigen Akteur in zukünftigen Fahrzeugführungskonzepten beizubehalten, nicht außer Acht lässt.

⁷ Winner (2015): Quo vadis, FAS? a: S. 1170; b: S. 1171

⁸ hierzu seien auch folgende Zitate genannt: „*For OEMs and other automotive industry players, the race is on for a share of this lucrative but still young market. Those who slow down may never catch up.*“ nach Viereckl et al. (2014): Racing Ahead - The Connected C@r 2014 Study, S. 10; Und: „*Norbert Reithofer [...] warnt seit Langem, dass es den Autoherstellern ergehen könnte wie den Produzenten von Schreibmaschinen oder Fotoapparaten, die vom Markt verschwanden oder bis zur Unkenntlichkeit schrumpften, als sich neue Technologien durchsetzten.*“ nach Hawranek & Schulz (2014): Die zweite Erfindung des Automobils, S. 16.

In den folgenden Abschnitten werden zunächst die Bedeutung funktionaler Architekturen im Kontext des Entwicklungsprozesses automatisierter Fahrzeuge konkretisiert und der Forschungsbedarf für die vorliegende Arbeit identifiziert.

1.2 Grundlagen zur Systemarchitektur

1.2.1 Begriffsdefinitionen

Gemäß Vogel et al.^{9a} existieren in der Fachliteratur eine Vielzahl unterschiedlicher Definitionen im Diskursbereich der Systemarchitektur. In der vorliegenden Arbeit werden die Definitionen von Vogel et al. übernommen, da diese die häufig genutzte Definition von Bass et al.¹⁰ aufgreifen und als zusätzliches Merkmal auch die Architektur als Disziplin beschrieben. Somit definiert sich der Begriff Systemarchitektur wie folgt^{9a}:

Die Systemarchitektur als strukturbeschreibendes Element:

„Die Systemarchitektur eines Systems beschreibt dessen Struktur respektive dessen Strukturen, dessen Bausteine (Software- und Hardware-Bausteine) sowie deren sichtbare Eigenschaften und Beziehungen sowohl zueinander als auch zu ihrer Umwelt.“

Die Systemarchitektur als Disziplin:

„Systemarchitektur als Disziplin befasst sich mit den architektonischen Tätigkeiten und den hiermit verbundenen Entscheidungen zum Entwurf und zur Umsetzung einer Systemarchitektur.“

Anhand der obigen Beschreibung lässt sich eine allgemeine Veranschaulichung des Begriffes „System“ ableiten, wie in Abbildung 1-1 gezeigt wird. Demnach ist ein System aus einem oder mehreren Software- sowie Hardware-Bausteinen aggregiert, wobei eine Aggregation eine „Ganzes-Teil-Hierarchie“ beschreibt¹¹. Software- und Hardware-Bausteine sind wiederum aus jeweils mehreren Software- und Hardware-Bausteinen zusammengesetzt und können hierarchisch aufgebaut sein^{9b}.

⁹ Vogel et al. (2009): Software-Architektur. a: S. 51; b: S. 58

¹⁰ Bass et al. (2003): Software Architecture in Practice, S. 3

¹¹ Weillkiens (2008): Systems Engineering mit SysML/UML, S. 215

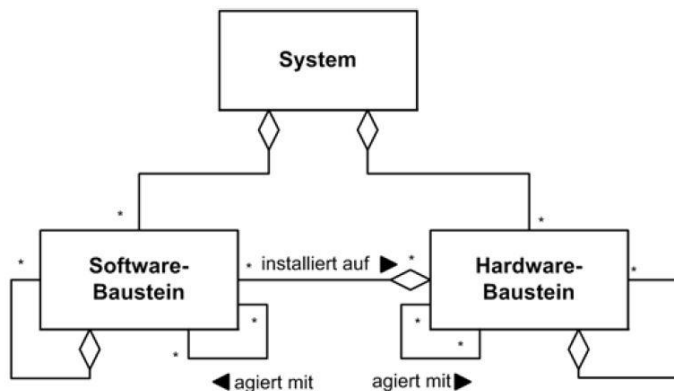


Abbildung 1-1: Bausteine eines Systems (in UML¹²-Notation) nach Vogel et al.¹³

Im Kontext eines automatisierten Fahrzeugführungssystems können als Beispiele für Hardware-Bausteine die Steuergeräte oder Computer genannt werden, auf denen die Software-Bausteine installiert sind. Je nach Definition der Systemgrenze zählen auch umfelderfassende Sensoren, Fahrdynamik-Sensoren oder Aktoren zur Beeinflussung der Fahrzeugquer- und Längsdynamik dazu. Software-Bausteine können einerseits Ausführungsumgebungen wie z. B. Betriebssysteme sein, die sog. Dienste zur Erfüllung nicht-funktionaler Anforderungen, z. B. zum Ressourcenmanagement oder zur Schnittstellenkommunikation, zur Verfügung stellen^{14a}. Andererseits werden durch Software-Bausteine auch die eigentlichen funktionalen Softwaremodule, z. B. ausführbare Programme, repräsentiert^{14b}.

Anhand Abbildung 1-1 wird ersichtlich, dass die Systemarchitektur die Software- sowie Hardware-Architektur mit einschließt. Die Definition dieser Begriffe nach Vogel et al. ist identisch mit den bereits oben genannten Definitionen zur Systemarchitektur, wenn der Begriff „System“ durch „Software“ bzw. „Hardware“ ersetzt wird, weshalb sie an dieser Stelle nicht wiederholt wird.

In Analogie zur Architektur eines Gebäudes, die in Form verschiedener Bauzeichnungen wie z. B. Grundrisse, Strom- oder Sanitärinstallationsplänen dokumentiert wird, existieren zur Beschreibung eines softwareintensiven Systems verschiedene sogenannter *Sichten*. Im Bereich der Software-Architektur empfiehlt der IEEE-Standard 1471¹⁵, diese Sichten systematisch an den Bedürfnissen der verschiedenen Interessensgruppen („Stakeholder“) auszurichten. Eine Übersicht über in der Praxis häufig genutzten

¹² UML: Unified Modeling Language. Für weiterführende Informationen siehe Weilkiens (2008): Systems Engineering mit SysML/UML

¹³ aus Vogel et al. (2009): Software-Architektur, S. 47

¹⁴ Vogel et al. (2009): Software-Architektur. a: S. 68; b: S. 48

¹⁵ IEEE (2000): Standard 1471-2000: Recommended practice for architectural description...

Sichtenmodelle findet sich in Vogel et al.¹⁶, wobei diese im Folgenden kurz beschrieben werden^{17 18}:

Die *Kontext-Sicht* beschreibt den Zusammenhang seines Systems mit seiner Umgebung (z. B. dessen Nutzer).

Die *Bausteinsicht* (häufig auch *logische Sicht* genannt) beschreibt die wichtigsten Systembausteine und ihre Interaktionen, die zur Erfüllung der funktionalen Anforderungen notwendig sind.

Die *Laufzeitsicht* (oder auch *Prozesssicht*) dokumentiert das dynamische Verhalten des Systems zur Laufzeit.

Die *Verteilungssicht* beschreibt, wie die Softwarebausteine auf die physikalischen Hardwarebausteine verteilt (partitioniert) werden.

In Bezug auf die vorliegende Arbeit ist die Bausteinsicht als wichtigste Dokumentationsform anzusehen, die auch häufig von anderen Literaturquellen zur Beschreibung der Systemarchitektur im Bereich des automatisierten Fahrens genutzt wird (vgl. Kap. 3). In der Literatur werden in diesem Zusammenhang ebenfalls sehr oft die Begriffe *logische* oder *funktionale Systemarchitektur* verwendet, zumeist ohne zugehörige Definition. Schäuffele & Zurawka schreiben dazu¹⁹:

„Die logische Systemarchitektur beschreibt eine abstrakte Lösung, vermeidet aber eine Festlegung auf eine konkrete technische Systemarchitektur. Es wird definiert, was das System leisten wird, aber nicht, wie es konkret realisiert wird. Es entsteht sozusagen ein abstraktes, logisches Modell des Systems und seiner Funktionen. [...] Grafische Notationen, wie Blockdiagramme oder Zustandsautomaten, eignen sich für eine modellbasierte Darstellung.“

In einem gemeinsamen Workshop zur Fahrzeugsystemtechnik unter Beteiligung des Fachgebietes Fahrzeugtechnik der TU Darmstadt, dem Institut für Regelungstechnik der TU Braunschweig und dem Lehrstuhl für Kraftfahrzeugtechnik der TU Dresden wurde der Begriff der funktionalen Systemarchitektur diskutiert²⁰. Demnach beschreibt die *funktionale Systemarchitektur* die Struktur sowie die Abhängigkeiten (Schnittstellen) zwischen Strukturelementen auf funktionaler Ebene, wobei keine Verknüpfung zu Hardwareelementen erfolgt. Sie beschreibt eine abstrakte Lösung, unabhängig von

¹⁶ Vogel et al. (2009): Software-Architektur, S. 94-99

¹⁷ Starke (2009): Effektive Software-Architekturen, S. 16

¹⁸ vgl. auch Darms (2007): Eine Basis-Systemarchitektur zur Sensordatenfusion..., S. 4

¹⁹ Schäuffele & Zurawka (2012): Automotive Software Engineering, S. 148

²⁰ Protokoll der Ergebnisse des 3. Workshop Fahrzeugsystemtechnik vom 24.-26.09.2014 in Radebeul, Dresden

einem konkreten technischen Lösungskonzept (wie z. B. die Eingrenzung auf einen speziellen Algorithmus)²¹.

Bei Vergleich der beiden, unabhängig voneinander erstellten Definitionen fällt auf, dass diese bedeutungsgleich sind. Aus diesem Grund werden im Rahmen der vorliegenden Arbeit die beiden Begriffe *logische* und *funktionale Systemarchitektur* als synonym eingestuft und der Begriff der funktionalen Systemarchitektur weiterverwendet.

1.2.2 Zweck der Systemarchitektur

Eine der wichtigsten Aufgaben der Systemarchitektur wurde bereits im vorherigen Abschnitt genannt: Mit Hilfe der im Architekturentwicklungsprozess entstehenden Sichten werden gezielt für die jeweiligen Interessensgruppen relevante Informationen über das System aufbereitet und transportiert. Diese Informationen können in verschiedenen Granularitätsstufen zur Verfügung gestellt werden. Demnach kann für die Projektleitung bzw. ein Steuerungskomitee eine vereinfachte Form der Bausteinsicht in Form einer Folienpräsentation (sog. *Pseudoarchitektur*) zweckmäßig sein, den prinzipiellen Aufbau des zu entwickelnden Systems zu diskutieren^{22a}. Ein Softwareentwickler hingegen benötigt eine detailliertere Beschreibung des von ihm zu erstellenden Softwaremoduls, z. B. in Form einer ausführlichen Bausteinsicht in UML-Notation inklusive einer Modul-Schnittstellenbeschreibung. Die Architektur stellt folglich ein gemeinsames Vokabular und einen Ordnungsrahmen dar, um selbst komplexe Sachverhalte und Zusammenhänge, wie sie auch in einem automatisierten Fahrzeug vorherrschen, verständlich zu machen^{22b} ²³. Vogel et al. bezeichnen die Systemarchitektur deshalb auch als einen „Katalysator für erfolgreiche Zusammenarbeit im Team“^{22b}.

Im Kontext des Gesamtentwicklungsprozesses kann die Architekturentwicklung in den linksseitigen Ast des bekannten V-Modells²⁴ eingeordnet werden, das sich für eine top-down motivierte und strukturierte Herangehensweise bei der Entwicklung von komplexen technischen Systemen bewährt hat. Dabei stellt die Erstellung der funktionalen und technischen²⁵ Architektur den Zwischenschritt zwischen der Anforderungsdefinition und

²¹ vgl. dazu auch die bedeutungsgleiche Definition von Behere & Törngren, abgeleitet aus der ISO 26262: Behere & Törngren (2015): A Functional Architecture for Autonomous Driving

²² vgl. Vogel et al. (2009): Software-Architektur. a: S. 7-8; b: S. 30

²³ Bass et al. (2013): Software Architecture in Practice, S. 29

²⁴ S. Internetseite der Bundesregierung zum V-Modell: http://www.cio.bund.de/Web/DE/Architekturen-und-Standards/V-Modell-XT/vmodell_xt_node.html, letzter Zugriff am 02.10.16

²⁵ Die technische Architektur wird als Synonym zur Hardware-Architektur verstanden, siehe hierzu auch Maurer (2013): Automotive Systems Engineering: A Personal Perspective, S. 24

der Implementierungsphase dar^{26a 27}. Seit dem Erscheinen der ISO-Norm 26262^{28a} zum Nachweis der funktionalen Sicherheit von Straßenfahrzeugen ist das V-Modell und somit auch die Entwicklung der Systemarchitektur als zentrales Element im Test und in der Absicherung von elektrischen und elektronischen Systemen in Fahrzeugen verankert. Insbesondere Teil 3 (Konzeptphase), in der sog. *Items* definiert werden, auf die die Norm angewendet wird (z. B. eine bestimmte Funktion, ein Teilsystem oder eine Anordnung von Teilsystemen) setzt eine vorherige Kenntnis der Systemstruktur inklusive deren funktionalen und nichtfunktionalen Anforderungen voraus^{28b 29}.

Anhand des vorherigen Absatzes wird klar, dass durch Festlegung der funktionalen Architektur in einer frühen Entwicklungsphase bereits wichtige Design-Entscheidungen getroffen werden. Dies führt dazu, dass den beteiligten Entwicklern komplexitätsreduzierende Randbedingungen vorgegeben werden können und so deren Kreativität „kanalisiert“ werden kann^{30a}. Es besteht jedoch auch die Gefahr, dass falsche Entscheidungen im weiteren Projektverlauf zu Problemen führen können, die dann nur noch unter hohem Aufwand und Kosten zu beheben sind^{30b}. Insbesondere die unzureichende Berücksichtigung von funktionalen und qualitätsbestimmenden Anforderungen an das System bergen die Gefahr, dass im Zuge der Implementierung vom ursprünglichen Architekturkonzept abgewichen werden muss. Es entsteht im schlimmsten Fall ein „Flickwerk“ von Softwaremodulen, deren Weiterentwicklung und Wartung immer mehr Zeit in Anspruch nimmt. Vogel et al. bezeichnen ein solches System als „Big Ball of Mud“ oder „Kludge“³¹. Im nächsten Abschnitt werden die Phasen des Architektur-Entwurfsprozesses vorgestellt, mit dessen Hilfe die Gefahr der Erreichung eines solchen unerwünschten Zustandes reduziert werden soll.

1.2.3 Architektur-Entwurfsprozess

Das Vorgehen beim Entwurf einer Systemarchitektur wird vorwiegend in der Fachliteratur für die Informationstechnik (IT) beschrieben und ist an Starke³² sowie Posch et al.^{26b} angelehnt. Abbildung 1-2 zeigt die wichtigsten Phasen des Prozesses, die im Folgenden näher beschrieben werden.

²⁶ Posch et al. (2007): Basiswissen Softwarearchitektur. a: S. 7; S. 58 ff

²⁷ vgl. Maurer (2012): Entwurf und Test von Fahrerassistenzsystemen, S. 45

²⁸ ISO 26262-1:2011 (2011): Road Vehicles - Functional Safety. a: -; b: Part 3, Ch. 5.4

²⁹ vgl. Hillenbrand (2012): Funktionale Sicherheit nach ISO 26262..., S. 92-94

³⁰ Bass et al. (2013): Software Architecture in Practice. a: S. 25; b: S. 28

³¹ vgl. Vogel et al. (2009): Software-Architektur, S. 4

³² Starke (2009): Effektive Software-Architekturen, S. 33 ff

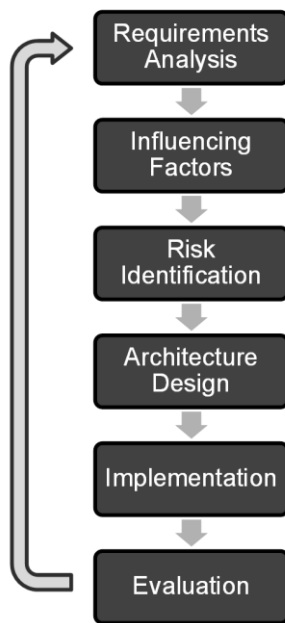


Abbildung 1-2: Allgemeiner Architekturentwurfsprozess³³

Im ersten Schritt erfolgt eine detaillierte Anforderungsanalyse an das zu entwickelnde System, wie sie auch im V-Modell gefordert wird. Ein zentrales Ergebnis stellt dabei die möglichst vollständige Auflistung der *funktionalen Anforderungen* dar. Bereits hier aber stößt der Systemarchitekt im Bereich der Fahrerassistenzsysteme und des automatisierten Fahrens auf das grundlegende Problem, dass eine vollständige (und textuelle) Anforderungsdefinition aufgrund der Vielzahl an möglichen Situationsparametern mit vertretbarem Aufwand a-priori häufig nicht möglich ist³⁴. Sehr oft werden aus diesem Grund im automotiven Bereich Anwendungsfälle in Form von Verkehrsszenarien³⁵ definiert, in denen das zu entwickelnde System aktiv sein wird und ein gewünschtes Verhalten zeigen soll. Diese werden in der Regel durch systematisches Variieren relevanter szenenbeschreibender Parameter generiert und in Form von Anwendungsfall-Katalogen dokumentiert³⁶. Die funktionalen Anforderungen für die in dieser Arbeit zu entwickelnden Referenzarchitektur werden in Kap. 2.2 dargestellt.

Von besonderer Relevanz für den Architektur-Entwurfsprozess sind die *nichtfunktionalen Anforderungen* an das System. Laut Vogel et al. werden nichtfunktionale Anforde-

³³ vgl. Lotz (2013): System Architectures for Automated Vehicle Guidance Concepts, S. 45

³⁴ vgl. Saust et al. (2009): Entwicklungsbegleitendes Simulations- und Testkonzept ..., S. 2

³⁵ Zur Abgrenzung der Begriffe „Szene“, „Situation“, „Szenario“ siehe Geyer et al. (2014): Concept and Development of a Unified Ontology...

³⁶ Beispiele hierfür sind u.a.: Geyer (2013): Entwicklung und Evaluierung eines kooperativen Interaktionskonzepts an Entscheidungspunkten...; Schuldt et al. (2014): Systematische Auswertung von Testfällen für Fahrfunktionen im modularen virtuellen Testbaukasten

rungen auch als *Qualitäten* oder *Qualitätsattribute* bezeichnet³⁷. Starke definiert die Qualität eines Systems analog als die Summe seiner nichtfunktionalen Eigenschaften, wobei die Architektur des Systems die Basis dafür bildet^{38a}. Eine detaillierte Übersicht über mögliche nichtfunktionale Anforderungen, respektive Qualitätsmerkmale, ist in Starke^{38b} gegeben, der sich auf die Norm ISO/IEC 9126³⁹ zur Sicherstellung von Softwarequalität bezieht. Beispiele für die Anwendung im Bereich des automatisierten Fahrens sind u.a. die Erweiterbarkeit des automatisierten Fahrzeugführungssystems auf neue Fahrumgebungen oder die möglichst einfache Testbarkeit des Gesamtsystems, wie bereits in Abschn. 1.1. erwähnt. Eine ausführliche Beschreibung der nichtfunktionalen Anforderungen im Rahmen der vorliegenden Arbeit erfolgt in Kap. 2.3.

Im zweiten Schritt des Architektur-Entwurfsprozesses werden Einflussfaktoren auf die Systemarchitektur identifiziert. Organisatorische Faktoren sind z. B. das für das Entwicklungsprojekt zu Grunde liegende Budget, der Zeitplan oder vorgegebene Entwicklungswerkzeuge und -Frameworks⁴⁰. Auch technische Einflussfaktoren, wie z. B. vorgegebene Hardware-Komponenten oder Datenschnittstellen, haben einen großen Einfluss auf die Systemarchitektur. Ebenfalls sollten projekt- bzw. unternehmenspolitische Einflussfaktoren nicht außer Acht gelassen werden, die eine Systemarchitektur oftmals stärker beeinflussen können als technische^{38c}.

Im weiteren Verlauf des Entwurfsprozesses müssen mögliche Risiken identifiziert werden, die sich aus dem Vergleich zwischen den bereits definierten Anforderungen und den im vorherigen Schritt gefundenen (den Lösungsraum einschränkenden) Einflussfaktoren ergeben. Als Folge werden daraufhin Strategien entwickelt, um die sich möglicherweise daraus entwickelnden negativen Konsequenzen zu minimieren. Etwaige Risiken können die mangelnde Verfügbarkeit projektrelevanter Ressourcen sein, wie z. B. eine geeignete Simulations- und Entwicklungsumgebung.

Im vierten Schritt des Architektur-Entwurfsprozesses erfolgt die eigentliche Entwicklung und Dokumentation der Systemarchitektur. Dazu ist zunächst eine Literaturrecherche notwendig, um ähnliche und auf das eigene Problem übertragbare *Architektur-Mittel* ausfindig zu machen. Abbildung 1-3 zeigt eine Übersicht über existierende Architektur-Mittel, die im Folgenden aufgrund der hohen Relevanz für die vorliegende Arbeit näher beschrieben werden.

³⁷ Vogel et al. (2009): Software-Architektur, S. 108

³⁸ Starke (2009): Effektive Software-Architekturen. a: S. 17; b: S. 57-58; c: S. 48-49

³⁹ ISO/IEC 9126 (2012): Software Engineering - Product Quality; Mittlerweile abgelöst durch die ISO/IEC 25000 (2014): Systems and Software Engineering...

⁴⁰ Posch et al. (2007): Basiswissen Softwarearchitektur, S. 75

Architektur-Prinzipien, auch *Architektur-Heuristiken* genannt, sind allgemeine Entwurfsprinzipien, die vor allem zwei Ziele verfolgen: Die Reduktion der Architektur-Komplexität sowie die Erhöhung der Flexibilität^{41a}. Somit unterstützen Sie den Architekten vor allem bei der Umsetzung nichtfunktionaler Anforderungen. Eines der wichtigsten Beispiele für Architekturprinzipien ist die *Modularität*, bei deren Anwendung ein System in Submodule zerlegt wird, die möglichst funktional gekapselt bzw. von anderen Modulen abgetrennt sind. Dabei vereinigt die Modularität die Architektur-Prinzipien der *Abstraktion*, des *Separation of Concerns*, sowie des *Information Hiding*^{41b}.

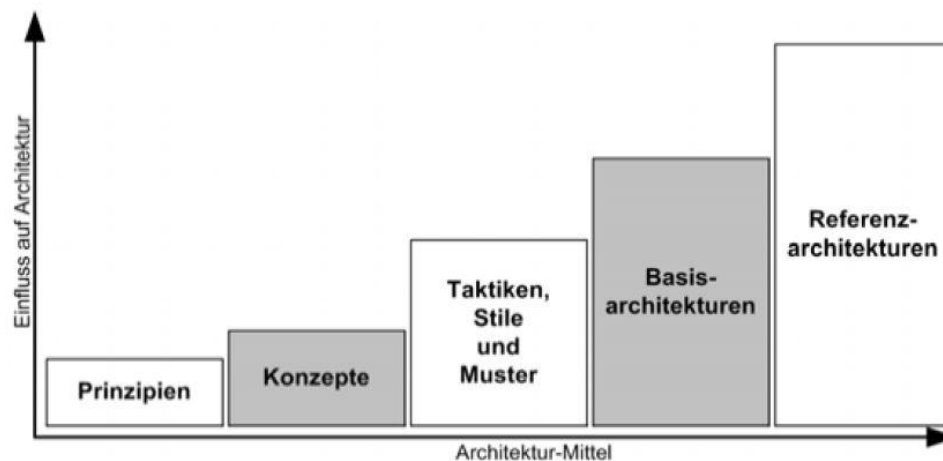


Abbildung 1-3: Einfluss von Architektur-Mitteln nach Vogel et al.⁴²

Architektur-Konzepte werden in Vogel et al. vorwiegend als Werkzeuge zur Umsetzung der Architektur auf Quellcode-Ebene beschrieben, z. B. mit Hilfe der Objektorientierung^{41c}.

Architektur-Muster (engl.: *Architecture patterns*) werden in der Literatur weitestgehend einheitlich als eine dreiteilige Regel beschrieben, „die die Beziehung zwischen einem bestimmten Kontext, einem Problem und einer Lösung ausdrückt“^{41d} 43. Dabei handelt es sich im Gegensatz zu Architektur-Prinzipien um konkrete Lösungen für wiederkehrende Probleme, oft in Form von UML-Diagrammen oder Quelltext-Fragmenten^{41e}. Beispiele für Architektur-Muster sind u.a. (Schnittstellen-) *Adapter* oder *Proxies*⁴⁴.

Basis-Architekturen stellen ein konkretes Architektur-Mittel dar, mit Hilfe dessen Systeme *ganzheitlich* strukturiert werden können. Ein Negativ-Beispiel für eine Basis-Architektur ist laut Vogel et al. der sogenannte *Monolith*, also die Zusammenfassung der

⁴¹ Vogel et al. (2009): Software-Architektur. a: S. 129; b: S. 145; c: S. 152 ff; d: S. 202; e: S. 205

⁴² entnommen aus Vogel et al. (2009): Software-Architektur, S. 127

⁴³ vgl. auch Bass et al. (2013): Software Architecture in Practice, S. 204

⁴⁴ vgl. Starke (2009): Effektive Software-Architekturen, S. 167 ff

Architektur in einen einzelnen Systembaustein^{45a}. Im Gegensatz dazu stehen u.a. *Schichtenarchitekturen*, die Systembausteine, die ähnliche Funktionen oder Verantwortlichkeiten besitzen, in dieselbe *Schicht* (engl.: *Layer*) eingruppiert^{45b}. Beispiele für Schichtenarchitekturen im Kontext des automatisierten Fahrens werden in Kap. 3.3 diskutiert.

Die bisher vorgestellten Architektur-Mittel sind Domänen-unabhängig und lassen sich deshalb fachübergreifend anwenden. *Referenzarchitekturen* hingegen berücksichtigen gezielt auch funktionale Anforderungen und stellen architektonische Gesamtlösungen für einen bestimmten Problembereich dar^{45c}. Ein charakteristisches Merkmal von Referenzarchitekturen ist, dass sie die Verteilung der Systemfunktionen, die sich aus den funktionalen Anforderungen ergeben, auf die einzelnen Systembausteine beschreiben und damit einhergehend auch deren Verantwortlichkeiten und Zusammenspiel festlegen. Definitionsgemäß spielt zu diesem Zweck die Bausteinsicht (vgl. Abschn. 1.2.1) als Dokumentationsform eine wichtige Rolle.

Im vorletzten Schritt des Architektur-Entwurfsprozesses erfolgt die Implementierung der Systemarchitektur. Dabei hat es sich bewährt, die definierte Struktur zunächst als sog. *Skelettsystem* umzusetzen, in der die einzelnen (Software-) Module in der zugrundeliegenden Entwicklungsumgebung erzeugt und bereits mit den notwendigen Schnittstellen ausgestattet werden^{45d}. Die eigentlichen Funktionalitäten werden dann im Zuge der Implementierung des Gesamtsystems gefüllt, wobei zunächst die Funktion nur eines spezifischen Anwendungsfalles vollständig umgesetzt wird (der sog. *Durchstich*) und die Umsetzung der anderen schrittweise erfolgt. Dies hat den Vorteil, dass in einem frühen Entwicklungsstadium bereits ein funktionierendes System zur Verfügung steht und somit frühzeitig Risiken identifiziert werden können^{45e}.

Auf Basis dieses Vorgehens ist es nun im letzten Schritt des Architektur-Entwurfsprozesses möglich, die implementierte Architektur bezogen auf die Erfüllung der zu Beginn definierten Anforderungen zu evaluieren. Da sich die Projektanforderungen innerhalb der Entwicklung ändern können, wird der Architektur-Entwurfsprozess in der Literatur als ein iterativer Prozess beschrieben⁴⁶, weshalb auch ein Rekursionspfeil in Abbildung 1-2 abgebildet ist. Grundlegende Architektur-Bewertungsverfahren werden im folgenden Abschnitt vorgestellt.

⁴⁵ Vogel et al. (2009): Software-Architektur. a: S. 216; b: S. 217; c: S. 254; d: S. 409; e: S. 400

⁴⁶ vgl. Starke (2009): Effektive Software-Architekturen, S. 25

1.2.4 Bewertungsverfahren von Systemarchitekturen

Charakteristisch für iterative Verfahren ist die Beurteilung, ob das Ergebnis des aktuellen Iterationsschrittes, in diesem Falle der Architektur-Entwurf, „gut genug“ ist und damit der Prozess abgeschlossen werden kann. Dies führt zur grundlegenden Frage: Was zeichnet eine gute Architektur aus und gibt es ein Kriterium, mit Hilfe dessen die Qualität des Architekturentwurfs bemessen werden kann?

Laut Starke^{47a} gibt es prinzipiell zwei Möglichkeiten, Softwareprojekte zu bewerten. Zum einen können organisatorische Aspekte im Entwicklungsprozess evaluiert werden, wie z. B. der dafür notwendige Ressourceneinsatz. Davon allerdings kausal auf die Qualität des Endproduktes zu schließen, ist nicht möglich. Die zweite Möglichkeit stellt die Bewertung sogenannter *Artefakte* dar, die im Laufe des Entwicklungsprozesses entstehen, wie z. B. Anforderungslisten, die bereits beschriebenen Architektur-Sichten oder der Quelltext. Nur wenige dieser Artefakte können zu einer quantitativen Bewertung der Architektur herangezogen werden. So lassen sich z. B. die Anzahl der Zeilen des Quelltextes, der notwendige Speicherbedarf oder die Anzahl der implementierten Funktionen pro Zeiteinheit theoretisch als Bewertungskriterium heranziehen. Bei genauerer Betrachtung quantitativer Bewertungsmetriken stellt sich aber heraus, dass diese in der Praxis erst zu einem sehr späten Entwicklungsstadium angewendet werden können und sie zudem strukturelle Schwachstellen des Architekturentwurfs möglicherweise nicht aufdecken können^{47b}. Starke kommt daher zur Schlussfolgerung, dass sich Systemarchitekturen nur qualitativ bewerten lassen^{47c}.

Eine ausführliche Übersicht über Bewertungsmethoden für Softwarearchitekturen geben Eicker et al.⁴⁸: Sie identifizieren insgesamt 18 verschiedene Verfahren, die sich grundsätzlich in zwei Kategorien einteilen lassen. Ein Teil davon sind die bereits erwähnten quantitativen Metriken, die ein bereits implementiertes System benötigen. In die zweite Kategorie werden Verfahren eingeordnet, die eine qualitative und szenariobasierte Bewertung beschreiben. Eine der verbreitetsten Verfahren davon ist die *Architecture Tradeoff Analysis Method* (ATAM), in der im ersten Schritt für das Projekt relevante Qualitätsmerkmale nach ihrer Wichtigkeit sortiert und in einem sog. Qualitätsbaum („utility tree“) abgelegt werden^{47d}. Für jedes der Qualitätsmerkmale, die auf den nicht-funktionalen Anforderungen basieren⁴⁹, wird im Anschluss mindestens ein konkretes Szenario definiert, mit Hilfe dessen das Merkmal untersucht werden kann. Eine detaillierte Übersicht über diese sog. *Qualitätsattributsszenarien* ist in Bass et al. gegeben⁵⁰,

⁴⁷ Starke (2009): Effektive Software-Architekturen. a: S. 301 ff; b: S. 303; c: S. 302; d: S. 309

⁴⁸ Eicker et al. (2007): Auswahl von Bewertungsmethoden für Softwarearchitekturen

⁴⁹ vgl. Abschnitt zu nichtfunktionalen Anforderungen in Kap. 1.2.3

⁵⁰ Bass et al. (2013): Software Architecture in Practice, S. 61 - 270

wie auch das Beispiel zum Qualitätsattribut der Änderbarkeit in Abbildung 1-4, das vom System fordert, eine Änderung der Nutzerschnittstelle (UI: User Interface) in drei Arbeitsstunden bewältigen zu können.

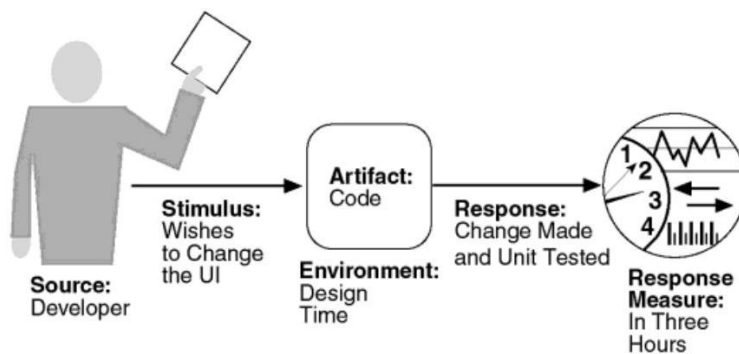


Abbildung 1-4: Beispiel für ein Qualitätsattributsszenario bezogen auf die Änderbarkeit⁵¹

Im weiteren Verlauf der ATAM wird diskutiert, wie die Architektur die Umsetzung der Szenarien begünstigt. Das Ergebnis der Methode ist ein schriftliches Protokoll bzw. eine Präsentation, aus der hervorgeht, wie gut die Szenarien von der Architektur unterstützt werden („Sensitivity“), welche qualitätsbezogenen Kompromisse eingegangen wurden („Tradeoff“) und welche Risiken sich daraus ergeben („Risk“). Die ATAM ist durch die ausführliche Prozessbeschreibung⁵² und der Beteiligung aller relevanten Stakeholder ein sehr ressourcen- und arbeitsintensives Verfahren^{53 54}.

Eine etwas leichter anzuwendende Methode ist die *Software Architecture Analysis Method* (SAAM), die in Abowd et al. beschrieben ist⁵⁵. Auch hier erfolgt eine Bewertung mit Hilfe von Qualitätsattributsszenarien, jedoch erfolgt keine Aufstellung eines priorisierten Qualitätsbaumes und keine detaillierte Sensitivity- und Tradeoff-Analyse. Stattdessen können mit dieser Methode mehrere Architekturvarianten schnell auf ihre Tauglichkeit hin überprüft werden, wobei als Ergebnis eine qualitative Aussage entsteht, ob eine Architekturvariante das Szenario prinzipiell unterstützt oder nicht⁵⁶. Weitere Architekturbewertungsverfahren dienen häufig einem sehr spezifischem Zweck, z. B. der Erreichung von wirtschaftlichen Geschäftszielen, und werden aufgrund der geringen Übertragbarkeit auf Architekturen für automatisiertes Fahren nicht weiter aufgeführt.

⁵¹ entnommen aus Bass et al. (2013): Software Architecture in Practice, S. 120

⁵² siehe Kazman et al. (2000): ATAM: Method for Architecture Evaluation

⁵³ Bass et al. (2013): Software Architecture in Practice, S. 415

⁵⁴ Eicker et al. (2007): Auswahl von Bewertungsmethoden für Softwarearchitekturen, S. 7

⁵⁵ Abowd et al. (2007): SAAM: A Method for Analyzing the Properties of Software Architectures

⁵⁶ siehe Beispiel zum Qualitätsattribut der Erweiterbarkeit in Abowd et al. (2007): SAAM: A Method for Analyzing the Properties of Software Architectures

Zusammenfassend kann festgehalten werden, dass in der Fachliteratur keine einheitliche Definition für eine „gute“ Architektur gefunden werden kann. Bass et al. sagen sogar:

“There is no such thing as an inherently good or bad architecture. Architectures are either more or less fit some purpose”^{57a}.

Dennoch scheint Einigkeit darin zu bestehen, dass sich die Qualität einer Architektur im Erfüllungsgrad der nichtfunktionalen Anforderungen widerspiegelt, was sich vor allem aus den vorgestellten Bewertungsverfahren ablesen lässt. Die Qualität stellt dabei kein absolutes Maß dar, sondern eine Güte im Hinblick auf spezielle Kriterien⁵⁸.

Dementsprechend kommt der Definition der nichtfunktionalen Anforderungen im Architekturentwicklungsprozess eine besondere Bedeutung zu: Eine Übererfüllung für das Projekt nicht relevanter Qualitätskriterien führt - genauso wie eine Untererfüllung - zu einem ineffizienten Entwicklungsprozess. Als Beispiel hierfür könnte eine Architektur sein, die eine hohe Änderbarkeit besitzt, aber absehbar in nur einem einzigen Prototypen-System zum Einsatz kommt^{57b}. Die Erfüllung *funktionaler* Anforderungen scheint hingegen bei der Architekturbewertung keine Rolle zu spielen, woraus sich schließen lässt, dass diese schlicht vorausgesetzt wird.

1.3 Entscheidungsverfahren komplexer Probleme

Im vorherigen Kapitel wurde der Architekturentwurfsprozess erläutert und ein Überblick gegeben, wie die daraus entstandene Architektur bewertet werden kann. Was jedoch aus der Fachliteratur nicht eindeutig hervorgeht ist, wie der Systemarchitekt im vierten Schritt des Entwurfsprozesses, also der eigentlichen Architekturentwicklung (siehe Abbildung 1-2), im Detail vorgehen sollte. Im Rahmen dieses Prozessschrittes werden zwar aus der Literatur Basis- oder Referenzarchitekturen ermittelt, jedoch erfüllen diese nicht zwangsläufig die Anforderungen des eigenen Entwicklungsprojektes und sind somit ggf. nicht unmittelbar übertragbar.

Ziel dieses Abschnittes ist es, herauszufinden, welche Entscheidungsverfahren ein Systemarchitekt beim eigentlichen Entwurf der Architektur einsetzen kann. Dazu ist es zunächst notwendig, das zugrunde liegende Problem näher zu analysieren. Gemäß

⁵⁷ Bass et al. (2013): Software Architecture in Practice. a: S. 19; b: S. 20

⁵⁸ Starke (2009): Effektive Software-Architekturen, S. 305

Grüning & Kühn^{59a} handelt es sich bei einem Entscheidungsproblem um ein *komplexes Problem*, wenn^{59b} ...

- dafür eine sehr große Anzahl an möglichen Problemlösungsvarianten existiert, die sich aus einer großen Anzahl von Entscheidungsvariablen und deren Ausprägungen ergeben.
- der Entscheider⁶⁰ (in diesem Falle der Systemarchitekt) nur partiell über die Erfahrung verfügt, die Konsequenzen der Problemlösungsvarianten bestimmen zu können.
- die Problemstruktur *schlecht strukturiert* ist, d.h. insbesondere, dass die Problembeschreibung nicht ausschließlich aus quantitativen Aspekten besteht.
- es sich beim vorliegenden Problem um ein *Gestaltungsproblem* (Design-Problem) und nicht um *Wahl-Problem* (Choice-Problem) handelt.

Alle vier dieser Aspekte treffen für den Bereich der Architekturentwicklung, zumindest für die Domäne der Fahrerassistenzsysteme und automatisierten Fahrzeuge, zu. Von besonderer Relevanz ist hierbei der Umstand, dass weder die Anforderungen an die Architektur noch die Bewertung, wie im vorherigen Kapitel ersichtlich wurde, rein quantitativ beschrieben werden können und deswegen ein *schlecht strukturiertes* und *schlecht definiertes* Problem vorliegt^{59c}, das somit definitionsgemäß als *komplex* bezeichnet werden kann^{59d}.

Neben der reinen Begriffsdefinition hilft die o.g. Problem-Charakterisierung dabei, die richtigen zugehörigen Entscheidungsverfahren zu identifizieren. Analytische Entscheidungsverfahren scheiden gemäß Grüning & Kühn bei komplexen und schlecht strukturierten Problemen als Lösungsmöglichkeit aus, sodass heuristische Entscheidungsverfahren eingesetzt werden müssen^{59e}, die im folgenden Zitat erläutert werden:

„Unter einer Heuristik wird [...] eine Denkregel verstanden, mit deren Hilfe sich der Aufwand zum Auffinden einer Lösung in komplexen Problemen reduzieren lässt. Dem Vorteil eines geringeren Lösungsaufwandes steht der Nachteil gegenüber, dass bloss eine befriedigende und nicht die optimale Lösung angestrebt werden kann. Es kann sogar vorkommen, dass überhaupt keine brauchbare Lösung aus der Anwendung der Heuristik resultiert“^{59f}.

Dieser zunächst ernüchternd erscheinenden Erkenntnis wird in der Praxis ein möglichst systematisch-rationaler Problemlösungsansatz entgegengestellt, um die potenziellen negativen Konsequenzen eines heuristischen Ansatzes abzumildern. Dabei sollte der Entscheidungsprozess, der zu einer Lösung führt, drei Anforderungen genügen: Er muss

⁵⁹ Grüning & Kühn (2013): Entscheidungsverfahren für komplexe Probleme. a: -; b: S. 7 ff; c: S. 50; d: S. 12; e: S. 50; f: S. 47

⁶⁰ Anmerkung: Der Entscheider wird in Grüning & Kühn als „Aktor“ bezeichnet.

durchgängig zielgerichtet sein, die Überlegungen sollten möglichst auf vollständigen und objektiven Informationen beruhen sowie einen systematischen und methodischen Charakter aufweisen, sodass er für Nichtbeteiligte nachvollziehbar ist^{61a}. Eines der wichtigsten heuristischen Prinzipien stellt die *Problemfaktorisierung* dar. Sie fordert, dass ein (komplexes) Gesamtproblem in eine Menge von lösbaren Teilproblemen zerlegt wird, die sequenziell (-hierarchisch) und / oder parallel angeordnet werden können^{61b}. In welcher Art und Weise die Teilprobleme aufgespalten und durchlaufen werden, muss gemäß der Grundform des allgemeinen heuristischen Entscheidungsverfahrens nach Grüning & Kühn zu Beginn des Lösungsprozesses mit Hilfe einer Problemanalyse geklärt werden. Im weiteren Verlauf werden für jedes der identifizierten Teilprobleme möglichst systematisch Lösungsvarianten erarbeitet und auf ihre Konsequenz in Bezug auf definierte Entscheidungskriterien bewertet^{61c}.

Das beschriebene Vorgehen ist keinesfalls neu und kommt in Form sog. *spezieller heuristischer Verfahren* ebenfalls in der technischen Produktentwicklung zum Einsatz. Dabei sind u.a. die *Portfolioanalyse* oder der *morphologische Kasten* bekannte Mittel, die zur Lösung der o.g. Einzel-Prozessschritte angewendet werden können^{61d}. Für den Entwurf von funktionalen Systemarchitekturen, die definitionsgemäß abstrakt und unabhängig von technischen Lösungskonzepten sind (vgl. Kap. 1.2.1), können diese speziellen Verfahren jedoch nicht direkt übernommen werden. Somit muss im Rahmen der vorliegenden Arbeit ein neues heuristisches Verfahren definiert und angewendet werden. Diesem Umstand kommt zugute, dass sich die im vorherigen Abschnitt vorgestellten Architekturbewertungsverfahren aufgrund ihrer Ähnlichkeit gut mit einem heuristischen Entscheidungsprozess verknüpfen lassen. Darüber hinaus kann auf Basis der Erkenntnisse dieses Abschnittes der in Abbildung 1-2 beschriebene Architektur-Entwurfsprozess durch das Hinzufügen der Prozessschritte der Problemfaktorisierung und Variantenbildung erweitert und detailliert werden, worauf in Kap. 1.5 (Methodik und Aufbau der Arbeit) näher eingegangen wird.

1.4 Ziele der Arbeit

Der Hintergrund der vorliegenden Arbeit stellt die Entwicklung einer funktionalen Systemarchitektur für das Forschungsprojekt PRORETA 3 dar. Ziel des Projekts ist es, die bisher wenig vernetzten und zumeist funktional unabhängigen Fahrerassistenzsystem-Funktionen des heutigen Standes der Technik in einen integralen Ansatz zur aktiven Sicherheit zusammenzuführen und - darauf aufbauend - eine manöverbasierte Fahr-

⁶¹ Grüning & Kühn (2013): Entscheidungsverfahren für komplexe Probleme. a: S. 38; S. 75; c: S. 66 ff; d: S. 51 und S. 106

zeugautomatisierungsfunktion zu realisieren (siehe dazu auch Kap. 2.1). Im Zuge dieser Tätigkeit wurde der Architektur-Entwurfsprozess, wie er in Kap. 1.2.3 beschrieben ist, durchlaufen und der Stand der Technik zu Basis- und Referenzarchitekturen für Fahrerassistenzsysteme und automatisierte Fahrzeuge ermittelt.

Dabei fällt auf, dass die Bedeutung sowie eine Diskussion über die Beschaffenheit von Systemarchitekturen aus wissenschaftlicher Sicht deutlich hinter der Beschreibung und Erreichung von funktionalen Zielen der Systeme zurückfallen. Insbesondere die Architektur als Disziplin, wie sie in Kap. 1.2.1 definiert wurde, wird im Kontext des automatisierten Fahrens kaum thematisiert, da die *Entscheidungen* einschließlich der Diskussion und Bewertung von *Entscheidungsalternativen*⁶², die zum jeweiligen Architekturentwurf führten, nur äußerst selten dokumentiert sind. Kazman et al. bestätigen diesen Eindruck:

„Developers tend to concentrate on the functional features of their architectures, and seldom address the ways in which their architectures support quality concerns within the system development life cycle”⁶³.

Dies führt dazu, dass Entwickler in der Literatur zwar Basis- und Referenzarchitekturen vorfinden, jedoch nur wenige oder keine Kriterien haben, die für ihr Problem bestgeeignete Variante zu bestimmen. Folglich müssen Systemarchitekten oftmals auf Basis unvollständiger Informationen und mit dem „Blick von außen“ abschätzen, ob eine gefundene Referenzarchitektur die funktionalen Anforderungen an das eigene Projekt adressiert und falls ja, ob und wie wichtige Qualitätsanforderungen darin umgesetzt werden. Hertzberg et al.^{64a} fassen das Problem für die Disziplin der Robotik sehr griffig zusammen:

„Eine geeignete Roboterkontrollarchitektur zu finden, ist ein schwieriges Problem. Und zwar ist es allein schon aus funktionaler Sicht schwierig, wenn also allein die einzelnen Roboterfunktionen zu einem funktionierenden Ganzen zusammengesetzt werden sollen; andere übliche Fragestellungen der Softwaretechnik, wie die Senkung der Lebenszykluskosten, sind dabei noch ganz außen vor. Für dieses schwierige Problem gibt es derzeit keine befriedigende Lösung. Das liegt nicht nur am Kaliber des Problems an sich, sondern auch daran, dass das Thema Roboterkontrollarchitekturen wissenschaftlich und technisch ein „sperriges“ Thema ist“^{64b}.

⁶² Die Betrachtung von Alternativen und die Dokumentation der Entscheidung machen dabei die Architektur für Außenstehende nachvollziehbar, vgl. Starke (2009): Effektive Software-Architekturen, S. 138-139

⁶³ Abowd et al. (2007): SAAM: A Method for Analyzing the Properties of Software Architectures, S. 1

⁶⁴ Hertzberg et al. (2012): Mobile Roboter. a: -; b: S. 318

Speziell bei der Betrachtung des Standes der Technik von Systemarchitekturen im Bereich der Fahrzeugautomatisierung und Robotik fällt darüber hinaus auf, dass sehr häufig sog. *hybride Architekturen* mit drei (Hierarchie-) Ebenen als Basis-Architekturen Verwendung finden (vgl. Kap. 3.3). Hertzberg et al. schreiben hierzu:

„Es herrscht weitgehend Einigkeit, dass hybride Architekturen eine adäquate Struktur für Roboterkontrollsoftware darstellen; dabei muss eine hybride Architektur nicht notwendigerweise genau drei Stufen haben, sondern es gibt Varianten von Zwischenstufen und Interaktionsprotokollen zwischen den Stufen, die auf zwei, vier, oder andere Stufenzahlen hinauslaufen [...]. Die richtige hybride Roboterkontrollarchitektur zu finden, ist derzeit ein offenes Forschungsproblem“⁶⁵.

Basierend auf den Ausführungen dieses Abschnittes ergeben sich die Ziele der vorliegenden Arbeit somit wie folgt:

- Es soll eine Referenzarchitektur in Form einer funktionalen Architektur entwickelt werden, die den funktionalen Anforderungen an ein automatisiertes Fahrzeugführungssystem genügt, das unterschiedliche Automationsgrade im Spektrum vom assistierten bis zum vollautomatisierten Fahren ermöglicht⁶⁶.
- Die Entscheidungen, die im Zuge des Architektur-Entwurfsprozesses getroffen werden, sollen dokumentiert werden. Insbesondere sollen dabei Entscheidungsalternativen erarbeitet und miteinander verglichen werden.
- Beim Vergleich zwischen und der Auswahl von Entscheidungsalternativen soll als Kriterium berücksichtigt werden, inwiefern die Alternativen die Erfüllung von nichtfunktionalen Anforderungen und somit der Qualitätsmerkmale begünstigen.

Mit Hilfe dieser Anforderungen soll die vorliegende Arbeit einen wissenschaftlichen Beitrag für die Disziplin der Fahrzeugsystemtechnik bieten, indem sie erstmals den *Erkenntnisweg*, der zur Entstehung einer Referenzarchitektur führt, über einen Top-Down-Ansatz ausführlich beschreibt. Dabei besteht keinesfalls der Anspruch auf Optimalität des Ergebnisses, da gemäß den Ausführungen der Kap. 1.2.4 und 1.3 weder ein Verfahren zur Entwicklung eines optimalen Architekturentwurfs noch eine Bewertungsmetrik zur Validierung dieses Anspruches existiert. Vielmehr soll dem Leser durch die strukturierte Vorgehensweise die Nachvollziehbarkeit der Entscheidungen und somit die Prüfung der Übertragbarkeit der Referenzarchitektur auf das eigene Problem erleichtert werden. Im nächsten Abschnitt erfolgt eine Konkretisierung der Methodik, um dieses Ziel zu erreichen.

⁶⁵ Hertzberg et al. (2012): Mobile Roboter, S. 325 - 326

⁶⁶ die funktionalen Anforderungen werden in Kap. 2.2 genauer spezifiziert

1.5 Methodik und Aufbau der Arbeit

Die Methodik der vorliegenden Arbeit kombiniert den in Abschn. 1.2.3 vorgestellten Architekturentwurfsprozess mit Elementen der heuristischen Problemlösungsmethodik aus Abschn. 1.3. Abbildung 1-5 veranschaulicht das Ergebnis.

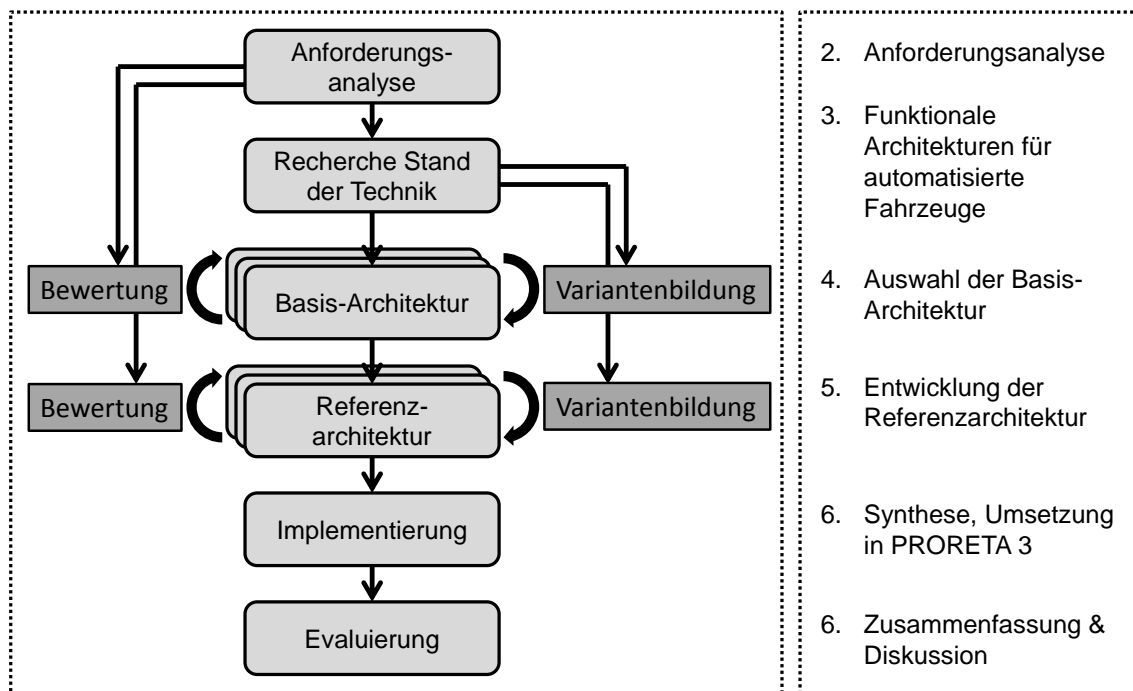


Abbildung 1-5: Methodik und Aufbau der Arbeit

Im ersten Schritt und in Kap. 2 (Anforderungsanalyse) werden die funktionalen und nichtfunktionalen Anforderungen an die Architektur spezifiziert. Die funktionalen Anforderungen stellen dabei, wie im Bereich der Fahrerassistenzsystementwicklung üblich, konkrete Verkehrsszenarien dar, die das System bewältigen können muss.

Im zweiten Schritt des Entwurfsprozesses erfolgt eine ausführliche Recherche zur Ermittlung des Standes der Forschung und Technik von Basis- und Referenzarchitekturen aus den Domänen der Fahrerassistenz, Fahrzeugautomatisierung und Robotik. Das Ergebnis dieses Arbeitsschrittes wird in Kap. 3 dargestellt.

Im Vergleich zum in Abbildung 1-2 beschriebenen Architekturentwurfsprozess fällt auf, dass die beiden Schritte der Identifikation der Einflussfaktoren und Risiken entfallen. Dies begründet sich dadurch, dass diese beiden Schritte hochgradig abhängig von den spezifischen Projektgegebenheiten sind und deswegen aufgrund des mangelnden Anspruchs auf Allgemeingültigkeit nicht Kern der vorliegenden Arbeit sind. Die projektspezifischen und technischen Randbedingungen des Projekts PRORETA 3 werden jedoch in Kap. 6.3 kurz erläutert.

Im weiteren Verlauf der Arbeit findet die Problemfaktorisierung als Kernelement des heuristischen Entscheidungsverfahrens Einzug in die hier vorgestellte Methodik. Dabei wird das Gesamtproblem des Entwurfs der Referenzarchitektur in sequenziell lösbare Teilprobleme aufgespalten, wobei im Verlauf des Prozesses eine schrittweise Entwicklung der Architektur vom Grob- bis zur Feinentwurf erfolgt. Für jeden Konkretisierungsschritt werden eine möglichst den Lösungsraum vollständig abdeckende Anzahl an Entwurfsvarianten gebildet, gegeneinander bewertet und schließlich als Ausgangsentwurf für den nächsten Konkretisierungsschritt ausgewählt. Diese Vorgehensweise besitzt drei Vorteile:

- Sie unterstützt die Forderung nach Bildung und Bewertung von Entwurfsvarianten⁶⁷.
- Durch die Hierarchisierung der Teilprobleme entfällt die aufwändige Ausarbeitung und Bewertung von Feinentwürfen, deren Grobstruktur bereits nicht den Anforderungen entsprechen⁶⁸.
- Durch die Variantenauswahl wird aus dem Gestaltungsproblem zum Teil ein Wahl-Problem (Choice-Problem), was die Gesamtkomplexität des Entwurfsprozesses reduziert (vgl. Kap. 1.3).

Dabei wird die Variantenbildung durch das Ergebnis der Literaturrecherche unterstützt, wie z. B. bei der Auswahl der Basis-Architektur in Kap. 4, bei der sich grundlegende Kontrollarchitekturen aus dem Bereich der Robotik zur Strukturierung eignen. Die Auswahl der Varianten erfolgt analog zu den in Kap. 1.2.4 beschriebenen Grundsätzen zur Bewertung von Systemarchitekturen. Zunächst wird identifiziert, ob sich die Varianten zur Umsetzung der funktionalen Anforderungen am Beispiel der zuvor definierten Verkehrsszenarien eignen. Falls nicht, kann die Eignung der Variante folglich falsifiziert werden und wird nicht weiter betrachtet. Übrigbleibende Varianten werden hinsichtlich ihrer Konsequenzen zur Erfüllung der nichtfunktionalen Anforderungen hin überprüft. Auf die Anwendung quantitativer und analytischer *Entscheidungsmaximen*⁶⁹, wie z. B. eine Punktbewertung der Varianten mit Gewichtung der Kategorien wird dabei bewusst verzichtet. Stattdessen werden die Gesamtkonsequenzen summarisch und textuell zu-

⁶⁷ vgl. Starke (2009): Effektive Software-Architekturen, S. 138-139

⁶⁸ s. auch Bass et al. (2013), S. 398: „*The more alternatives, the more time could be spent evaluating them. Try to eliminate alternatives quickly so that the number of viable potential alternatives is small.*”

⁶⁹ Eine Übersicht über analytische Entscheidungsmaximen ist in Grüning & Kühn (2013): Entscheidungsverfahren für komplexe Probleme, S. 159 gegeben

sammengefasst⁷⁰, analog zu der in Kap. 1.2.4 vorgestellten Architektur-Bewertungsmethode SAAM. Als Ergebnis des Prozessschrittes wird die zu entwickelnde Referenzarchitektur einschließlich der Partitionierung der Systemfunktionen auf einzelne Systembausteine in Kap. 6.1 beschrieben.

Im letzten Schritt der vorliegenden Arbeit und in Kap. 6.3 wird die exemplarische Anwendung der entwickelten Architektur im Forschungsprojekt PRORETA 3 einschließlich der Implementierung in ein Versuchsfahrzeug demonstriert und diskutiert.

⁷⁰ vgl. auch Grüning & Kühn (2013), S. 168: „Die Autoren haben persönlich mit der summarischen Vorgehensweise positive Erfahrungen gemacht. In der Praxis führt eine Abwägung aller in einer Entscheidungsmatrix zusammengefassten Konsequenzen zu bewussteren und klarer begründeten Entscheidungen als der Rückgriff auf rechnerisch ermittelte und häufig schwer verständliche Gesamtkonsequenzen.“

2 Anforderungsanalyse

In diesem Kapitel wird zunächst das Forschungsprojekt PRORETA 3 vorgestellt, das den projektspezifischen Rahmen der zugrundeliegenden Arbeit darstellt. Ausgehend davon werden allgemeine funktionale Anforderungen identifiziert, die eine Systemarchitektur für ein automatisiertes Fahrzeug mit Fahrereinbindung erfüllen muss. Schließlich werden ebenfalls nichtfunktionale Anforderungen definiert, um eine Grundlage zur Bewertung der Güte verschiedener Architekturvarianten zu schaffen.

2.1 Das Projekt PRORETA 3

Beim Forschungsprojekt PRORETA 3 handelt es sich um ein Kooperationsprojekt zwischen der Continental AG und der TU Darmstadt⁷¹. Das Projekt fand von April 2011 bis September 2014 unter Beteiligung der Fachgebiete Fahrzeugtechnik, Regelungsmethoden und Robotik, Regelungstechnik und Mechatronik sowie dem Institut für Arbeitswissenschaft statt.

Ziel des Projektes war zum einen, die heute im Fahrzeug erhältlichen Fahrerwarnsysteme und Assistenzsysteme zur Unfallvermeidung bzw. Unfallfolgenminderung mit aktivem Eingriff in die Fahrzeuglängs- und Querdynamik in einem konsistenten Gesamtansatz zusammenzufassen. Der Hintergrund dieser Forderung entstammt dem Wunsch, einer Entwicklungstendenz des „linearen Hinzufügens“ von Assistenzfunktionen und deren Steuergeräte in bestehende Systeme entgegenzutreten⁷² und somit auch die Gefahr konkurrierender Verhaltensmuster (z. B. die Entscheidung zwischen Notausweichen, Notbremsen, und kombiniertem Ausweichen und Bremsen) sowie der Fahrerüberforderung durch Zunahme der Informations- und Bedienschnittstellen⁷³ zu mindern. Im entsprechenden Assistenzmodus *Sicherheitskorridor*, der als Paradigma für einen sichereren und kollisionsfreien Fahrbereich steht, fährt der Fahrer wie gewohnt selbst. Falls das Assistenzsystem im Fahrbetrieb die Notwendigkeit eines Eingriffes in die Längs- oder Querdynamik des Fahrzeuges feststellt, um z. B. eine Kollision mit anderen Ver-

⁷¹ Eine Übersicht über die Historie des Projektes inkl. der Ergebnisse der Vorgängerprojekte PRORETA 1 und 2 findet sich auf der Internetseite www.proreta.de (letzter Zugriff am 02.10.16)

⁷² vgl. Reichart & Bielefeld (2012): Einflüsse von Fahrerassistenzsystemen auf die Systemarchitektur im Kraftfahrzeug, S. 84

⁷³ vgl. Kauer et al. (2010): How to conduct a car?, S. 1214

kehrsubjekten zu vermeiden, so entspricht dies dem Erreichen der Grenzen des *Sicherheitskorridors*, was dem Fahrer im Vorfeld durch zielgerichtete Warnelemente visueller, akustischer und kinästhetischer Natur mitgeteilt wird.

Darüber hinaus bestand das Entwicklungsziel, dem Fahrer einen zweiten Assistenzmodus anzubieten, die sog. *kooperative Automation*. Unter dem Begriff wird im Rahmen dieser Arbeit eine intensive und kooperative Interaktion zwischen Fahrer und dem automatisierten Fahrzeugführungssystem verstanden, mit dem Ziel, auf Basis gegenseitigem Informierens, Empfehlens und Befürwortens die Fahraufgabe gemeinsam effektiver bewältigen zu können⁷⁴. Dabei konzentriert sich PRORETA auf eine spezifische Ausprägung der *kooperativen Automation*, nämlich der *Manöverdelegation*, in dem der Fahrer sequenziell verschiedene Fahrmanöver, wie z. B. Fahrstreifenwechsel oder Abbiegemanöver, an das Fahrzeug delegiert und somit gemäß des 3-Ebenen Modells nach Donges⁷⁵ von der Fahrzeugstabilisierungsaufgabe entlastet wird (s. auch Kap. 5.1.1). Das System verfolgt die Grundidee des an der TU Darmstadt entwickelten *Conduct-by-Wire*-Konzepts (CbW), dessen Kern die manöverbasierte Fahrzeugautomatisierung enthält⁷⁶ (vgl. Kap. 3.4.4). Die *kooperative Automation* lässt sich in der von der Bundesanstalt für Straßenwesen herausgegebenen Klassifikation von Automatisierungsgraden⁷⁷ in die *teilautomatisierte Fahrzeugführung* eingliedern⁷⁸, bei der das System die Quer- und Längsführung für einen gewissen Zeitraum und / oder in spezifischen Situationen übernimmt, der Fahrer dies aber dauerhaft überwachen und jederzeit für eine Rückübernahme bereit sein muss. Die Integration des Fahrers besitzt dabei den Vorteil, dass die teilweise heute noch vorliegenden technologischen Defizite umfelderfassender Assistenzsysteme, z. B. hinsichtlich der Situationsanalyse und Entscheidungsfindung, durch Einbringung der menschlichen Erfahrung kompensiert werden können, was sich in einer geringeren Gesamtsystemkomplexität niederschlägt⁷⁹.

Im Rahmen des Projektes wurden die beschriebenen Funktionen in ein Versuchsfahrzeug implementiert. Somit konnte weltweit erstmals ein Konzept der manöverbasierten Fahrzeugführung in Realfahrten erprobt werden (siehe Kap. 6.3). Die in den nächsten Abschnitten beschriebenen Anforderungen basieren teilweise auf der Zielsetzung von

⁷⁴ Definition basierend auf Lotz (2013): System Architectures for Automated Vehicle Guidance Concepts, S. 40

⁷⁵ vgl. Donges (2015): Fahrerverhaltensmodelle

⁷⁶ vgl. Franz et al. (2015): Conduct-by-Wire

⁷⁷ siehe Gasser et al. (2012): Rechtsfolgen zunehmender Fahrzeugautomatisierung

⁷⁸ Geyer (2013): Entwicklung und Evaluierung eines kooperativen Interaktionskonzepts..., S. 6-8

⁷⁹ vgl. Ibanez Guzman et al. (2012): Autonomous Driving: Context and State of the Art, S. 1304

PRORETA 3, werden jedoch verallgemeinert und auf die vollautomatisierte Fahrzeugführung erweitert.

2.2 Funktionale Anforderungen

2.2.1 Anforderungsdefinition in der Fahrerassistenzsystem-Entwicklung

Funktionale Anforderungen an ein technisches System sind im Allgemeinen erwünschte oder geforderte *Fähigkeiten* des Systems, die es in einem definierten *Nutzungskontext* zur Erfüllung einer definierten Aufgabe bzw. zum Erreichen eines definierten *Nutzungsziels* besitzen muss⁸⁰.

Klassische *Nutzungsziele* bei der Entwicklung von Fahrerassistenz- oder Fahrzeugautomatisierungssystemen stellen dabei Assistenzziele dar, von denen sich die Entwickler eine Erhöhung der Sicherheit und / oder des Komforts versprechen und werden häufig durch Defizite der menschlichen Leistungsfähigkeit motiviert, z. B. wie in Vollrath et al.⁸¹ durch eine detaillierte Analyse des Unfallgeschehens und der zugrunde gelegten Fahrer-Fehlhandlungen. Zur Erreichung der Nutzungsziele besitzt das System dafür notwendige *Fähigkeiten*: Siedersberger⁸² unterscheidet bei automatisierten Fahrzeugen zwischen Fähigkeiten zur Wahrnehmung, zur Entscheidung und zur Aktionsausführung, wobei insbesondere letztere für die Anforderungsdefinition von Bedeutung sind und im Folgenden zunächst als *Verhaltensweisen* bezeichnet werden⁸³. Der *Nutzungskontext* von Assistenz- und Automatisierungssystemen (diese werden im nachfolgenden als *A/A-Systeme* bezeichnet) ist der Straßenverkehr; Diese einfach formulierte Aussage birgt jedoch eine enorme Komplexität in der Entwicklung und Absicherung solcher Systeme, was an folgendem Beispiel veranschaulicht wird: Während das Assistenzziel und die gewünschte Verhaltensweise eines Frontkollisionsschutzsystems vergleichsweise einfach formuliert werden können, nämlich das Einleiten einer automatischen Notbremsung zur Vermeidung eines Unfalles im Längsverkehr (Auffahrunfall), so vielfältig stellt sich der Nutzungskontext dar^{84 85}. Entsprechend soll das Notbremssys-

⁸⁰ Eigene Definition

⁸¹ Vollrath et al. (2006): Ableitung von Anforderungen an Fahrerassistenzsysteme...

⁸² Siedersberger (2003): Komponenten zur automatischen Fahrzeugführung..., S. 73 ff

⁸³ Siedersberger (2003) diskutiert die Begriffe "Fähigkeiten" und "Verhaltensfähigkeiten" bzw. „Verhaltensweisen“ noch detaillierter. Aus Gründen der Verständlichkeit werden in diesem Kapitel Fähigkeiten zur Aktionsausführung als Verhaltensweisen bezeichnet. Beispiele hierfür sind u.a. Fahrmanöver.

⁸⁴ vgl. Winner (2015): Grundlagen von Frontkollisionsschutzsystemen

tem idealerweise in verschiedenen Verkehrsdomänen (z. B. Autobahn, Landstraße), in verschiedenen Szenarien (ein- oder zweistreifige Richtungsfahrbahnen, Kreuzungen, verschiedene Fahrbahn-Beschaffenheiten) und bei unterschiedlichen Verkehrsobjekt-Konstellationen (Ausgangsabstand, Geschwindigkeiten, Verzögerungen) einsetzbar sein. Folglich sind weniger die eigentlichen Nutzungsziele eines Assistenzsystems, sondern vielmehr der Nutzungskontext ausschlaggebend für die Komplexität der Anforderungsermittlung⁸⁶.

Aufgrund der Vielzahl dieser kontextbeschreibenden Parameter hat sich bei der Entwicklung von A/A-Systemen eine Verkehrsszenarien-basierte Anforderungsdefinition bewährt, da eine vollständige (textuelle) Anforderungsdefinition, wie beispielsweise in Form einer Anforderungsliste, mit vertretbarem Aufwand häufig nicht möglich ist⁸⁷. In der Literatur finden sich als Ergebnis dieses Prozesses deshalb umfangreiche Kataloge oder Listen, die unter verschiedenen Namen geführt werden, z. B. Anwendungsfall- bzw. Use-Case-Kataloge oder Situations- bzw. Szenarienkataloge^{88 89}. Diese werden häufig durch eine systematische Kombination von situationsbeschreibenden Merkmalen und deren Ausprägungen gewonnen⁹⁰. Der Vorteil dieses Verfahrens liegt darin, dass prinzipiell alle möglichen Lösungen im Umfang des vorher durch Merkmalsdefinition diskretisierten Lösungsraumes gefunden werden. Der Nachteil liegt jedoch in der Tatsache, dass neben der hohen Anzahl an Szenarien auch bestimmte Kombination zu nicht sinnvollen Lösungen führen können, die im Anschluss manuell aussortiert werden müssen.

Wie bereits im Rahmen der Entwicklung der Anforderungen für das manöverbasierte Automationskonzept *Conduct-by-Wire* festgestellt wurde, eignet sich deshalb ein rein auf Kombinatorik beruhendes Anforderungsermittlungsverfahren nicht für eine ganzheitliche Systembetrachtung⁸⁸. Geyer⁸⁸ schlägt für diesen Fall eine Analyse bestehender Gesetze und Regelwerke vor, wobei nur entsprechend zulässige Kombinationen von Situationsparametern beachtet werden und so die Gesamtlösungsmenge auf ein handhabbares Maß reduziert wird. So lassen sich insbesondere aus den Richtlinien zur Anlage von Straßen viele Einschränkungen für die Gestaltung (statischer) Verkehrsszenarien

⁸⁵ vgl. Reschka et al. (2015): Entwicklungsprozess von Kollisionsschutzsystemen...

⁸⁶ vgl. dazu auch Omasreiter & Metzker (2004): A Context-Driven Use Case Creation Process...

⁸⁷ Saust et al. (2009): Entwicklungsbegleitendes Simulations- und Testkonzept..., S. 90

⁸⁸ Geyer (2013): Entwicklung und Evaluierung eines kooperativen Interaktionskonzepts..., S. 21-24

⁸⁹ Eine Abgrenzung der Begriffe „Szenerie“, „Situation“ und „Szenario“ erfolgt hier: Geyer et al. (2014): Concept and Development of a Unified Ontology...

⁹⁰ Ein anschauliches Beispiel dafür findet sich u.a. in Domsch & Negele (2008): Einsatz von Referenzfahrsituationen bei der Entwicklung von Fahrerassistenzsystemen

gewinnen. Dieser Ansatz wird für die Anforderungsermittlung in der vorliegenden Arbeit übernommen. Ein weiteres Werkzeug zur Reduktion des Anforderungsumfanges ist das Zusammenfassen vieler situationsbeschreibender Parameter auf wenige, möglichst voneinander unabhängige Merkmale⁹¹.

Die Herausforderung besteht nun darin, die für die Entwicklung einer Systemarchitektur relevanten Parameter zu identifizieren. Hierzu werden im Folgenden die beiden beschriebenen Ansätze kombiniert. Im ersten Schritt erfolgt eine strukturelle Analyse des Anwendungskontextes der Systemarchitektur für ein automatisiertes Fahrzeug auf Basis verschiedener Straßenkategorien und deren bauliche Entwurfs Elemente gemäß Straßenbaurichtlinien. Im zweiten Schritt werden für jede betrachtete Kategorie relevante *Verhaltensweisen* abgeleitet, die das automatisierte Fahrzeug zur Bewältigung einer Fahrmission für die genannte Kategorie besitzen muss. Werden diese vollständig ermittelt, so können im Anschluss an die Anforderungsdefinition die dafür notwendigen *Fähigkeiten zur Entscheidung* und *Wahrnehmung* schrittweise abgeleitet und konkretisiert werden. Somit erhält der Systemarchitekt eine weitestgehend vollständige Liste notwendiger System-Fähigkeiten, die als Funktionen auf die System- bzw. Softwarebausteine partitioniert werden können. Im nächsten Abschnitt wird die vorgestellte Methodik auf das deutsche Straßennetz angewendet.

2.2.2 Szenarienkatalog

Im Straßenwesen existieren zahlreiche technische Regelwerke, die die Planung, Konzeption und den Bau der Straßenverkehrs-Infrastruktur betreffen⁹². Von grundlegender Bedeutung sind dabei die folgenden Straßenbaurichtlinien, herausgegeben von der Forschungsgesellschaft für Straßen- und Verkehrswesen (FGSV):

- Richtlinie für die Anlage von Autobahnen (RAA)⁹³
- Richtlinie für die Anlage von Landstraßen (RAL)⁹⁴
- Richtlinie für die Anlage von Stadtstraßen (RASt)⁹⁵

Der Geltungsbereich der genannten Richtlinien decken den Lösungsraum des Straßennetzes, d.h. von Fernautobahnen bis hin zum Wohnweg, vollständig ab⁹⁶.

⁹¹ vgl. Weitzel et al. (2015): Absicherungsstrategien für Fahrerassistenzsysteme, S. 49

⁹² eine Übersicht der technischen Regelwerke ist zugänglich unter „Straßenbau A-Z: Sammlung technischer Regelwerke und Amtlicher Bestimmungen für das Straßenwesen“, Erich Schmitt Verlag, 2015; Internetpräsenz: <http://www.strassenbaudigital.de/>, letzter Zugriff am 03.10.16

⁹³ FGSV (2008): Richtlinie für die Anlage von Autobahnen

⁹⁴ FGSV (2012): Richtlinie für die Anlage von Landstraßen

⁹⁵ FGSV (2006): Richtlinie für die Anlage von Stadtstraßen

Durch Verknüpfung der Straßen verschiedener Kategorien entsteht ein Straßennetz. Dabei geht aus den Richtlinien eindeutig hervor, welche Kombination aus verknüpften Straßensegmenten und Verkehrsknoten zulässig ist.

Autobahn¹⁰⁰

Führt die Fahrmission des automatisierten Fahrzeugs über eine Autobahn, so müssen auf Basis der o.g. Strukturelemente verschiedene Verhaltensweisen des Systems beherrscht werden. Diese werden im Folgenden hergeleitet, wobei aus Gründen der Übersichtlichkeit in *Freifahrt*, also unbeeinflusstes Fahren ohne unmittelbare Nähe zu anderen Verkehrsteilnehmern, und in die *Fahrt mit Interaktion* mit anderen Verkehrsteilnehmern unterschieden wird.

⁹⁶ FGSV (2008): Richtlinie für die Anlage von Autobahnen, S. 7

⁹⁷ Kraftfahrstraßen sind durch das Zeichen 331 der StVO (Straßenverkehrsordnung) gekennzeichnet

⁹⁸ FGSV (2012): Richtlinie für die Anlage von Landstraßen, Blatt 2

⁹⁹ FGSV (2006): Richtlinie für die Anlage von Stadtstraßen, S. 13

27

gehenden möglichen Fahrmanöver variiert werden, solange, bis keine neuen Verhaltensweisen mehr gefunden werden können. Das Ergebnis dieses Arbeitsschrittes ist in Tabelle A-1 in Anhang A.1 dokumentiert. Für das knotenpunktfreie Befahren einer Autobahn in Freifahrt sind relevante Verhaltensweisen das *Fahrstreifen befahren*, der *Fahrstreifenwechsel* mit dem Parameter *links* bzw. *rechts* auf den in Querrichtung nächsten durchgehenden Fahrstreifen oder Zusatzfahrstreifen, das *Ein- und Ausfädeln* auf entsprechenden Ein- und Ausfädelungstreifen (bei Rastanlagen) sowie das *Halten* auf dem Seitenstreifen.

Im nächsten Schritt erfolgt die Betrachtung möglicher Interaktionen mit anderen Verkehrsobjekten. Szenarien mit Interaktion treten dann auf, wenn durch die geplante Verhaltensweise des automatisierten (Ego-) Fahrzeuges die Verhaltensweise eines anderen Verkehrsobjektes unmittelbar beeinflusst oder zur Durchführung einer anderen Verhaltensweise gezwungen wird und umgekehrt. Ein Beispiel für ein Szenario mit Interaktion ist das Einfädeln des Ego-Fahrzeuges von einem Einfädelungstreifen auf einen durchgehenden Fahrstreifen, wobei ein sich darauf befindendes Fahrzeug bewusst für einen Eingriff in die Längsdynamik (z. B. durch leichtes Abbremsen) entscheidet, um diesen Vorgang zu ermöglichen. Umgekehrt könnte sich das automatisierte Fahrzeug ebenfalls dazu entscheiden, einem anderen Fahrzeug das Einfädeln zu ermöglichen. Ein solches Verhalten wird in der Literatur häufig als *kooperatives Verhalten* bezeichnet. Düring & Pascheka bezeichnen ein Verhalten als kooperativ, wenn der Gesamtnutzen, d.h. die Summe der Einzelnutzen in einer Nutzenfunktion, von zwei oder mehr Agenten durch Kooperation mindestens genauso hoch oder höher ist als ohne eine Kooperation¹⁰¹. Da kooperatives Verhalten im Straßenverkehr notwendig ist und sogar durch den Gesetzgeber gefordert wird¹⁰², wird dieser Aspekt im Rahmen dieser Arbeit durch die Aufnahme entsprechender Szenarien zentral in den Anforderungen berücksichtigt (vgl. Tabelle A-1).

Knotenpunkte bei Autobahnen können entweder als Autobahnkreuze bzw. -Dreiecke (planfreie Knotenpunkte) oder als Anschlussstellen (teilplanfreie Knotenpunkte) gestaltet sein¹⁰³. Entsprechende bauliche Strukturelemente sind Ein- und Ausfahrten, verbindende Rampen und Verflechtungsbereiche. Passende Verhaltensweisen sind in Tabelle A-1 aufgeführt, wobei wieder die Manöver *Fahrstreifenwechsel* sowie *Ein- und Ausfädeln* maßgeblich sind.

¹⁰¹ Düring & Pascheka (2014): Cooperative decentralized decision making for conflict resolution...

¹⁰² In der Straßenverkehrsordnung (StVO) wird an mehreren Stellen auf kooperatives Verhalten verwiesen. Bereits in §1, Absatz 1 heißt es: „Die Teilnahme am Straßenverkehr erfordert ständige Vorsicht und gegenseitige Rücksicht.“ Eine weiterführende Diskussion dazu ist zu finden in: Graupner (2013): Bewertung des Standes der Forschung zum Kooperationsverhalten im Straßenverkehr, S. 11 ff

¹⁰³ vgl. Übersicht in: FGSV (2012): Richtlinie für die Anlage von Landstraßen, Blatt 32

Als Ergebnis dieser Arbeitsschritte steht nun ein Szenarienkatalog zur Verfügung, der die Verkehrsdomäne der Autobahn abdeckt. Nun können alle gefundenen Verhaltensweisen in einer Liste zusammengefasst und gruppiert werden, da diese von besonderem Interesse für die Architekturentwicklung sind. Verhaltensweisen mit einer hohen Ähnlichkeit werden dabei zusammengefasst: So werden die Verhalten Ein- bzw. Ausfädeln in das Verhalten Fahrstreifenwechsel überführt, da es sich dabei um einen Fahrstreifenwechsel auf einen Ausfädelungstreifen bzw. von einem Einfädelungstreifen auf einen durchgehenden Fahrstreifen handelt. Das Ergebnis dieses Arbeitsschrittes ist in Tabelle A-2 in Anhang A.1 aufgeführt. Es wurden demnach drei grundlegende Verhalten identifiziert: Das Befahren eines Fahrstreifens einschließlich des Einhaltens eines sicheren Abstandes zu einem evtl. unmittelbar vorausfahrenden Fahrzeug, der Fahrstreifenwechsel sowie das Halten auf einem Seitenstreifen. Mit speziellen Attributen können diese grundlegenden Verhalten detailliert werden. So besitzt der Fahrstreifenwechsel des Ego-Fahrzeuges, bei dem sich auf dem Zielfahrstreifen andere Verkehrsobjekte befinden, das Attribut *mit Interaktion*. Ein weiteres Attribut *unter Zwang* verdeutlicht in diesem Zusammenhang, dass ein Fahrstreifenwechsel unmittelbar und zwingend erforderlich ist, falls z. B. der eigene Fahrstreifen in Kürze endet. Diese Attribute sind voneinander unabhängig. So kann ein Fahrstreifenwechsel *mit Interaktion* und *unter Zwang* stattfinden, wenn z. B. das Ego-Fahrzeug den bald endenden Einfädelungstreifen verlassen und auf einen dicht befahrenen durchgehenden Fahrstreifen wechseln muss.

Landstraße¹⁰⁴

Landstraßen sind im Gegensatz zu Autobahnen einbahnig und in der Regel zweistreifig ausgeführt. Landstraßen einer höherrangigen können zusätzlich zu den durchgehenden Fahrstreifen Überholfahrstreifen besitzen, die in abwechselnder Fahrtrichtung angeordnet sind. Statt eines Seitenstreifens können Landstraßen Nothaltebuchten aufweisen. Planfreie Knotenpunkte ändern sich im Vergleich zur Autobahn nicht. Plangleiche Kreuzungen oder Einmündungen können mit oder ohne Lichtsignalanlagen geregelt sein. Gemäß RAL kann es an Kreuzungszufahrten ein oder zwei durchgehende Fahrstreifen sowie jeweils maximal einen Abbiege-Fahrstreifen (links / rechts) geben. Eine weitere (plangleiche) Knotenpunktart ist der Kreisverkehr, der über einen oder zwei Fahrstreifen innerhalb der Kreisbahn verfügen kann, jedoch in den Ein- Ausfahrten lediglich über einen Fahrstreifen verfügen sollte.

Aufgrund dieser Strukturelemente kommen weitere notwendige Verhaltensweisen in das Repertoire des automatisierten Fahrzeugs hinzu. Identifiziert wurden das *Abbiegen* an

¹⁰⁴ Alle Angaben zur Gestaltung von Strukturelementen sind direkt der RAL entnommen und werden im Weiteren nicht einzeln referenziert.

Kreuzungen¹⁰⁵ mit den Parametern *links* bzw. *rechts* sowie das *Kreuzen*, also das Geradeaus-Fahren, jeweils unter Beachtung der für den Knotenpunkt gültigen Vorfahrtsregelung. Diese Verhaltensweisen werden nicht als Interaktion oder Kooperation verstanden, da die entsprechenden Verhalten fest in der Straßenverkehrsordnung niedergeschrieben sind und somit kein Kooperationspielraum besteht. Das *Einfahren* in und *Ausfahren* aus Kreisverkehren werden als eigenständige Verhaltensweisen dokumentiert. Der Szenarien katalog für Landstraßen ist in Tabelle A-3 sowie in auf die Verhaltensweisen reduzierter Form in Tabelle A-4 in Anhang A.2 gegeben.

Stadtstraßen¹⁰⁶

Stadtstraßen innerhalb bebauter Gebiete sind in der Regel plangleich ausgeführt und besitzen im Gegensatz zu Autobahnen und Landstraßen zahlreiche Strukturelemente, die speziell für Fußgänger und Fahrradfahrer angelegt sind. Die konkrete bauliche Ausbildung von Kreuzungen ist weniger stark reglementiert. So gibt es beispielsweise keine Beschränkung über die Anzahl und nur wenige Beschränkungen zur Anordnung von Fahrstreifen. Je nach Bemessung des Verkehrsaufkommens an Kreuzungen kann mehr als ein Abbiege-Fahrstreifen je Richtung erforderlich sein¹⁰⁷. Dieser Umstand erfordert jedoch nicht grundlegende neue Verhaltensweisen, da die Manöver Abbiegen bzw. Kreuzen nach wie vor Bestand haben. Der Szenarien katalog für den Stadtverkehr ist in Tabelle A-5 in Anhang A.3 dokumentiert sowie die Liste der entsprechenden Verhaltensweisen in Tabelle A-6. Neu hinzugekommen ist das Verhalten *Wenden* an Wendefahrbahnen¹⁰⁸.

Generell herrscht innerorts eine starke Diversifikation der straßenbaulichen Elemente. Diese richten sich maßgeblich nach dem Nutzungsanspruch der jeweiligen Straßenkategorie. So besteht z. B. bei sog. Wohnwegen oder Wohnstraßen ein sog. *besonderer*

¹⁰⁵ Verkehrsrechtlich wird in Abbiegen und Einbiegen unterschieden. Abbiegen ist das Einfahren eines Fahrzeugs in eine untergeordnete Straße und Einbiegen das Einfahren in eine übergeordnete Straße. Da dies jedoch hauptsächlich für die vorherrschende Vorfahrtsregelung von Interesse ist, die ohnehin unabhängig davon berücksichtigt wird, wird im Rahmen dieser Arbeit das Einbiegen mit dem Abbiegen gleichgesetzt.

¹⁰⁶ Alle Angaben zur Gestaltung von Strukturelementen sind direkt der RAS entnommen und werden im Weiteren nicht einzeln referenziert.

¹⁰⁷ Die genaue Kreuzungsauslegung erfolgt hierbei mit Hilfe des Handbuchs für die Bemessung von Straßenverkehrsanlagen (HBS), herausgegeben von der FGSV, 2009

¹⁰⁸ Gemäß einem Gerichtsurteil des OLG Karlsruhe (VRS 60, 143) kann statt eines Wendevorgangs auch ein doppelter Linksabbiege-Vorgang vorliegen, wenn ein breiter Mittelstreifen vorhanden ist. Aus Gründen der Unterscheidbarkeit der Verhaltensweisen wird dies jedoch im Rahmen dieser Arbeit vernachlässigt.

Nutzungsanspruch hinsichtlich des Aufenthalts und des Parkens. Entsprechend finden sich hier häufig Bauelemente wie Parkflächen, Begrünung sowie Maßnahmen zur Verkehrsberuhigung. Szenarien dieser Art finden in der vorliegenden Arbeit keine Berücksichtigung, da diese Straßen nicht dem primären Nutzungsanspruchs der Fortbewegung entsprechen.

2.2.3 Zusammenfassung

Im Rahmen dieses Kapitels wurden notwendige Verhaltensweisen für ein automatisiertes Fahrzeug für die Domänen Autobahn, Land- und Stadtstraße systematisch identifiziert. Anhand der Ausführungen der vorherigen Abschnitte ist ersichtlich, dass grundlegende Verhaltensweisen existieren, die in mehreren dieser Domänen benötigt werden, wie z. B. der Fahrstreifenwechsel. Somit lassen sich diese weiter zusammenfassen. Tabelle 2-1 zeigt das Ergebnis. Darin werden die einzelnen Verhaltensweisen geordnet und beschrieben, einschließlich einer Übersicht, für welche Domäne die entsprechende Verhaltensweise relevant ist.

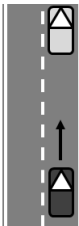
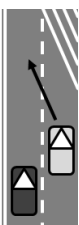
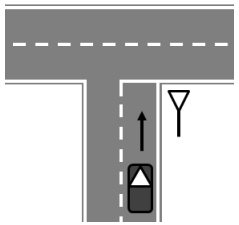
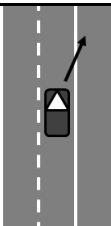
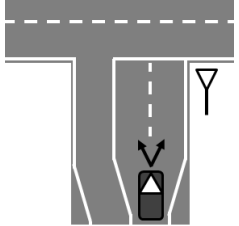

Die Zusammenfassung der Verhaltensweisen über verschiedene Domänengrenzen ist aus Architektur-Sicht dann zulässig, wenn sich die Verhaltensmechanismen und deren Übergänge nicht grundlegend ändern. So lassen sich z. B. Verkehrsobjekt-Konstellationen in verschiedenen Szenarien über Zeitmaße normieren, wie es z. B. bei Assistenzsystemen des Standes der Technik üblich ist. Zeitmaße, wie die Zeitlücke zum vorausfahrenden Fahrzeug beim Adaptive Cruise Control (ACC)¹⁰⁹ oder die Time-To-Collision (TTC) bei Frontkollisionsschutzsystemen¹¹⁰, erlauben somit eine Anwendung der Funktionen in verschiedenen Geschwindigkeitsbereichen und Domänen.

Sofern daraus resultierende Unterschiede seitens der Reichweite von umfelderfassenden Sensoren sowie der Reglerparametrierung notwendiger Fahrdynamikregler für die verschiedenen Domänen berücksichtigt werden, ist auch eine domänenunabhängige Betrachtung der gezeigten Verhaltensweisen im Rahmen der Architekturentwicklung möglich. Dies wird im weiteren Verlauf der Arbeit vorausgesetzt.

¹⁰⁹ vgl. Winner (2015): Adaptive Cruise Control

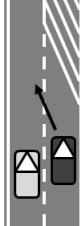
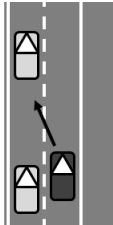
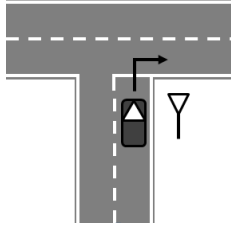
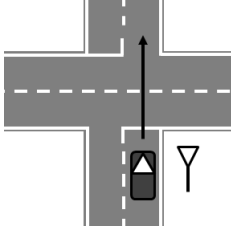
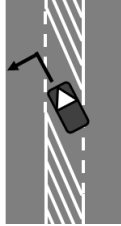
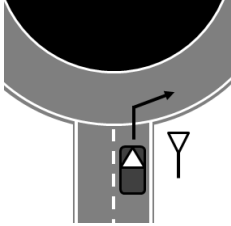
¹¹⁰ vgl. Winner (2015): Grundlagen von Frontkollisionsschutzsystemen

Tabelle 2-1: Übersicht der notwendigen Verhaltensweisen für automatisiertes Fahren¹¹¹

Nr.	Verhaltensweise	Parameter	Beschreibung	Relevanz			Schematische Abb.
				AB	LS	SS	
1	Fahrstreifen befahren	mit / ohne unmittelbar vorausfahrendes VO	Fahren auf einem FS (FS, Rampe, Kreisfahrbahn, Bypass etc.) mit Wunschgeschwindigkeit und ggf. unter Einhaltung eines sicheren Abstandes zum vorausfahrenden VO	X	X	X	
	Fahrstreifenwechsel zulassen	von VO links, von VO rechts	Fahren auf einem FS und anderes VO von links / rechts durch aktive Anpassung der Längsgeschwindigkeit auf eigenen FS wechseln lassen (auch Einfädelung)	X	X	X	
	Annähern an Knoten / Fußgängerüberweg		Dem plangleichen Knotenpunkt bzw. Fußgängerüberweg annähern und Längsgeschwindigkeit den Erfordernissen der Vorfahrtsregelung anpassen (ggf. verlangsamen, anhalten)		X	X	
2	Halten		Halten auf Seitenstreifen (Autobahn) oder in Nothaltebucht (Landstraße)	X	X		
3	In Fahrstreifen einordnen	links, rechts	An Kreuzungen in gewünschten FS einordnen, wenn keine durchgehende FS-Markierung vorhanden ist (i.d.R. bei Einbiegungen)		X	X	
4	Fahrstreifenwechsel (allgemein)	links, rechts	FS-Wechsel auf nächsten FS links / rechts des Ursprungs-FS (auch Ein- und Ausfädelungstreifen, Abbiege-FS etc.)	X	X	X	

¹¹¹ Abkürzungen: AB: Autobahn; LS: Landstraße; SS: Stadtstraße; VO: Verkehrsobjekt; FS: Fahrstreifen

Tabelle 2-1: Übersicht der notwendigen Verhaltensweisen für automatisiertes Fahren (Fortsetzung)

Nr.	Verhaltensweise	Parameter	Beschreibung	Relevanz			Schematische Abb.
				AB	LS	SS	
4ff	Fahrstreifenwechsel (unter Zwang)	links, rechts	FS-Wechsel aufgrund Ende des aktuellen FS (z. B. bei Einfädelungstreifen) oder um eine gewünschte Fahrroute einzuhalten (z. B. Erreichen eines Abbiege-FS)	X	X	X	
	Fahrstreifenwechsel (Interaktion)	links, rechts	FS-Wechsel auf nächsten FS in geeignete Lücke anderer VO auf dem Ziel-FS. An Kreuzungen ggf. unter Beachtung von Radfahrstreifen	X	X	X	
5	Abbiegen	links, rechts	An Kreuzungen abbiegen unter Beachtung der Vorfahrtsregelung		X	X	
6	Kreuzen		Kreuzung kreuzen (geradeaus fahren) unter Beachtung der Vorfahrtsregelung		X	X	
7	Wenden		Von Linksabbiege-FS oder Wendefahrbahn aus unter der Beachtung der Vorfahrtsregelung wenden			X	
8	In Kreisverkehr ein- bzw. ausfahren	Einfahren, Ausfahren	In Kreisverkehr unter Beachtung der Vorfahrtsregelung ein- bzw. ausfahren		X	X	

Bei Betrachtung der identifizierten Verhaltensweisen fällt ebenfalls auf, dass diese auf einer strukturierten Fahrumgebung aufbauen und ein strukturiertes, verkehrsregelkonformes Verhalten anderer Verkehrsteilnehmer voraussetzen. Entsprechend wird sich eine auf diesen Anforderungen basierende Architektur voraussichtlich nicht für das Fahren in unstrukturierten Umgebungen, wie z. B. Off-Road, Land- und Forstwirtschaftswegen oder Parkplätzen sowie in Situationen mit sehr hoher Verkehrsdichte, eignen.

Die funktionalen Anforderungen sind mit der reinen Aufstellung notwendiger Verhaltensweisen noch unvollständig. Weitere Anforderungen werden im Folgenden spezifiziert:

- **Fähigkeit zur Missionsplanung und -ausführung:** Bei vollautomatisierten Fahrzeugen gehen den gezeigten Verhaltensweisen Entscheidungen hinsichtlich der Missionsdurchführung, insbesondere der Navigation im Straßennetz, voraus (vgl. Kap. 5.1.1). Die Architektur muss entsprechend eine solche Funktion vorhalten.
- **Fähigkeit zur Verhaltensausführung:** Die gezeigten Verhaltensweisen müssen auf Fahrzeugstabilisierungsebene (vgl. Kap. 5.1.1) umgesetzt werden können. Dies erfordert die Fähigkeit, die Fahrzeuglängs- und Querdynamik des Fahrzeugs beeinflussen zu können.
- **Assistiertes (unfallvermeidendes) Fahren:** Gemäß gewünschtem Funktionsumfangs laut Kap. 1.4 und 2.1 soll das Fahrzeug im assistierten Modus konventionell steuerbar sein und eine integrale Sicherheitsfunktion bei drohenden Unfällen aktiv eingreifen. Auch hierzu lässt sich ein Szenarienkatalog erarbeiten. Im Rahmen der vorliegenden Arbeit wurde hierzu eine In-Depth-Unfallanalyse der Bundesanstalt für Straßenwesen ausgewertet¹¹² und die häufigsten Unfallszenarien schwerer Unfälle in Anhang A.4 in Tabelle A-7 abgelegt. Analog zur Zusammenfassung der Verhaltensweisen für automatisiertes Fahren können diese Szenarien in das grundlegende Verhalten *Unfallvermeidung* zusammengefasst werden. Eine Unterteilung in mögliche Verhaltensweisen wie z. B. „Notbremsen“ und „Notausweichen“ wird bewusst verzichtet, da die Entscheidung, welches dieser Verhalten zur Unfallvermeidung notwendig ist, von vielen Einflussfaktoren abhängt und nicht immer a-priori für jedes Szenario angegeben werden kann. Wichtig in diesem Zusammenhang ist, dass prinzipiell die Möglichkeit besteht, solche Manöver auszuführen. Darüber hinaus besteht die Notwendigkeit, den Fahrer über eine geeignete Mensch-Maschine-Schnittstelle (MMS) warnen bzw. Handlungsaufforderungen ausgeben zu können (vgl. Kap. 6.3).
- **Mensch-Maschine-Schnittstelle:** Neben der Notwendigkeit von Fahrerwarnungen besteht im automatisierten Betrieb die Forderung nach einer für die ver-

¹¹² Vollrath et al. (2006): Ableitung von Anforderungen an Fahrerassistenzsysteme...

schiedenen Automationsgrade geeigneten MMS, um mögliche Fahrmanöver oder den gewünschten Zielort eingeben zu können. Die Architektur muss aus diesem Grund eine solche Funktion unterstützen.

Im folgenden Abschnitt werden die nichtfunktionalen Anforderungen spezifiziert, die eine besonders hohe Relevanz hinsichtlich der Erreichung einer hohen Architekturqualität besitzen.

2.3 Nichtfunktionale Anforderungen

2.3.1 Allgemeine nichtfunktionale Anforderungen

Wie bereits in Kap. 1.2.4 (Bewertungsverfahren von Systemarchitekturen) beschrieben wurde, wird die Qualität einer Architektur anhand der Erfüllung der nichtfunktionalen Anforderungen bemessen, die deshalb auch als Qualitätsattribute bezeichnet werden¹¹³. In der Literatur werden diese Attribute in verschiedene Kategorien zusammengefasst, die im Folgenden kurz vorgestellt werden¹¹⁴:

- **Funktionalität:** Vorhandensein von Funktionen, die den funktionalen Anforderungen genügen, wobei angemessene und richtige Ergebnisse ausgegeben werden. Dabei werden anwendungsspezifische Normen und gesetzliche Bestimmungen eingehalten. Die Interoperabilität, d.h. die Einbettung in die Betriebsinfrastruktur, ist sichergestellt.
- **Sicherheit (security):** Das Softwareprodukt ist sicher (secure), d.h. vor unberechtigtem, versehentlichen und vorsätzlichem Zugriff geschützt.
- **Zuverlässigkeit:** Die Software kann das notwendige Leistungsniveau halten, besitzt eine geringe Versagenshäufigkeit und eine hohe Verfügbarkeit. Die Software ist robust gegenüber Fehlern oder der Nicht-Einhaltung ihrer spezifizierten Schnittstellen. Falsche Eingaben („garbage-in“) sollten nicht zu falschen Ausgaben („garbage-out“) führen¹¹⁵.
- **Benutzbarkeit:** Die Bedienung der Software ist für den Nutzer verständlich und einfach zu erlernen (Software-Ergonomie).
- **Effizienz:** Beschreibt das Verhältnis zwischen dem Leistungsniveau der Software und dem Umfang der eingesetzten Betriebsmittel.

¹¹³ Bass (2013): Software Architecture in Practice (s. Inhaltsverzeichnis)

¹¹⁴ auf Basis von Bass (2013): Software Architecture in Practice, Buchteil II sowie Starke (2009): Effektive Software-Architekturen, S. 57 ff

¹¹⁵ vgl. Starke (2009): Effektive Software-Architekturen, S. 154

- **Änderbarkeit:** Beschreibt den Aufwand, der zur Umsetzung von Änderungen notwendig ist, wobei Änderungen Fehlerkorrekturen, Verbesserungen sowie das Hinzufügen neuer (inkrementeller) funktionaler Anforderungen mit einschließen.
- **Testbarkeit:** Die Testbarkeit ist der Aufwand, in der Software enthaltene Fehler auffinden zu können und möglichst schnell eine hohe Testabdeckung zu erreichen.
- **Übertragbarkeit:** Beschreibt die Eignung der Software, in eine andere Hard- und Software-Umgebung übertragen zu werden.

Die genannten Qualitätsattribute beziehen sich dabei auf allgemeine Softwareprojekte und finden in der einschlägigen Fachliteratur häufig Anwendung im Bereich der Informationstechnologie. Für die Erstellung einer Systemarchitektur für assistiertes und automatisiertes Fahren können jedoch Funktionsdomänen-spezifische Schwerpunkte dieser Attribute auf Basis aktueller Entwicklungstendenzen abgeleitet werden, was im Folgenden aufgezeigt wird.

2.3.2 Nichtfunktionale Anforderungen für A/A-Systeme

Absicherung und Test

Von übergeordneter Bedeutung in der automotiven Softwareentwicklung ist die Testbarkeit, wie bereits in der Motivation dieser Arbeit deutlich wurde (vgl. Kap. 1.1). Verwandte Themengebiete sind in diesem Zusammenhang der Freigabeaufwand sowie der Nachweis der funktionalen Sicherheit nach ISO 26262.

Insbesondere bei Fahrerassistenzsystemen mit Umfeldwahrnehmung nimmt der Test- und Freigabeaufwand aufgrund der Systemkomplexität und der Vielzahl möglicher Kontext-beschreibender Parameter kontinuierlich zu. Aus diesem Grund entwickelte sich in der jüngsten Vergangenheit der Trend zur Virtualisierung des Testens. Dabei werden, abhängig vom aktuellen Entwicklungsschritt, das gesamte oder nur Teile des Systems in einer Computersimulation getestet und verifiziert. Die Vorteile dieses Vorgehens sind der hohe Grad an Reproduzierbarkeit sowie die Möglichkeit, das System oder dessen Subsysteme bereits zu einem sehr frühen Zeitpunkt (an dem beispielsweise noch kein realer Versuchsträger existiert) testen zu können¹¹⁶.

Ein weiterer wichtiger Aspekt bei Entwicklung und Freigabe ist die Arbeitsteilung, die in der Praxis über Unternehmensgrenzen hinweg geschieht. So werden Hard- und Software-Subsysteme sowie Komponenten nicht zwangsläufig aus einer Hand entwickelt,

¹¹⁶ s. Hakuli & Krug (2015): Virtuelle Integration, S. 126

sondern es findet vielmehr eine verteilte Entwicklung von OEMs und Zulieferern statt, wobei in der Regel eine gemeinsame Integrations- und Testumgebung eingesetzt wird¹¹⁷. In diesem Zusammenhang werden häufig Softwaremodule in Form einer „Black Box“ an Entwicklungspartner gegeben, um das eigene Know-How zu schützen.

Die Frage ist nun, wie sich die o.g. Anforderungen in konkrete, also überprüfbare, nicht-funktionale Anforderungen für die Systemarchitektur übersetzen lassen. Bezugnehmend auf die Testbarkeit schlagen Bass et al. zwei *Entwicklungstaktiken* vor¹¹⁸. Zum einen sollten die Softwarekomponenten beobachtbar und kontrollierbar sein, d.h. dass die internen und externen Systemzustände von außen vorgegeben und ausgelesen werden können, um so effektiv testen und Fehler finden zu können. In der objektorientierten Programmierung übliche Maßnahmen dafür stellen z. B. *set-* und *get-Methoden* dar, mit Hilfe derer auf interne Variablen zugegriffen werden können. Ebenfalls wird die Verwendung einer Testumgebung empfohlen, die ein reproduzierbares, wiederholtes Abspielen von Systemeingangssignalen und das Protokollieren der Systemausgangssignale und -Zustände ermöglicht. Diese Anforderungen beziehen sich jedoch nicht direkt auf die funktionale Architektur und die Partitionierung von Systemfunktionen auf Softwaremodule. Die zweite Taktik zur Erhöhung der Testbarkeit stellt laut Bass et al. die Reduktion der (strukturellen) Systemkomplexität dar. Hauptgrund dafür ist, dass komplexe Systeme einen größeren operativen Zustandsraum der Systemvariablen und deren Kombinationen besitzen, was es schwierig macht, einen exakten Zustand, z. B. zum Eingrenzen eines Fehlers, reproduzierbar wiederherzustellen. Methoden zur Verringerung der Komplexität sind¹¹⁹:

- Vermeiden zyklischer Abhängigkeiten, z. B. „Kreisbezüge“ von drei oder mehr Modulen¹²⁰
- Isolieren bzw. Kapseln von Abhängigkeiten des Systems zu dessen Umwelt (Systemgrenze)
- Generelle Reduktion von Abhängigkeiten zwischen den Komponenten
- Vermeiden der Notwendigkeit eines zu jederzeit vollständig konsistenten Datensatzes
- Vermeiden nichtdeterministischer Elemente in der Software, soweit möglich

Bass et al. bringen diese Methoden auch in direkten Zusammenhang mit den Architekturmitteln, die bereits in Abbildung 1-3 in Kap. 1.2.3 vorgestellt wurden. Architekturprinzipien, die in direktem Zusammenhang zur Testbarkeit stehen sind somit¹²¹:

¹¹⁷ vgl. Hakuli & Krug (2015): Virtuelle Integration, S. 127 ff

¹¹⁸ Bass et al. (2013): Software Architecture in Practice, S. 164 ff

¹¹⁹ auf Basis von Bass et al. (2013): Software Architecture in Practice, S. 167 - 168

¹²⁰ s. dazu auch Starke (2009): Effektive Software-Architekturen, S. 163

- *Prinzip der hohen Kohäsion:* Die Kohäsion beschreibt die Abhängigkeiten innerhalb eines Moduls. Eine hohe Kohäsion liegt dann vor, wenn ein Modul „alle die zum Verstehen und Ändern relevanter Eigenschaften in seiner Beschreibung vereint“¹²². Dies lässt sich u.a. durch Kapselung verwandter (funktionaler) Anforderungen in einem Modul erreichen, da somit ein hoher Kommunikationsbedarf innerhalb und nicht außerhalb des Moduls besteht. Gemäß Vogel et al. eignen sich Architekturen mit einer hohen Kohäsion sehr gut, um Systembausteine als Black Boxes zu repräsentieren.
- *Prinzip der losen Kopplung:* Module besitzen untereinander eine möglichst geringe (logische) äußere Abhängigkeit. Module können nicht auf interne Datenstrukturen anderer Module zugreifen, sondern nur über eine möglichst kleine Anzahl an Daten-Schnittstellen miteinander kommunizieren. Durch eine lose Kopplung kann ein Modul verändert werden, ohne dass ein tiefes Verständnis anderer Module vorausgesetzt wird. Eine lose Kopplung führt architektonisch in der Regel zu einer hohen Kohäsion.
- *Information-Hiding-Prinzip:* Dieses Prinzip ist stark mit dem Prinzip der losen Kopplung verwandt. Es besagt, dass gekoppelte Module nur die Informationen austauschen, die auch wirklich benötigt werden. Interne (private) Daten werden verborgen. Schichtenarchitekturen sind dann vorteilhaft, wenn eine Schicht nur auf die jeweils über- oder untergeordnete Schicht zugreifen kann und Hierarchieebenen nicht übersprungen werden.
- *Separation-of-Concerns-Prinzip:* Mit Hilfe dieses Prinzips werden Teilaspekte eines Problems aufgetrennt, sodass jedes unabhängig vom anderen behandelt werden kann. Häufig werden dabei die funktionalen Anforderungen eines Systems als Dekompositionskriterium angewendet, sodass jedes Subsystem eine Teilfunktion realisiert. Wichtig dabei ist, dass technische bzw. Hardware-Aspekte bei der Dekomposition keine Rolle spielen sollen¹²³. Das Separation-of-Concerns-Prinzip ist damit ein Haupttreiber für die Ermöglichung der Arbeitsteilung.

Neben den genannten Architekturprinzipien besitzen auch die Architekturmittel der Basis-Architekturen (vgl. Abbildung 1-3) Einfluss auf die Systemkomplexität und damit auf die Testbarkeit. So empfehlen Bass et al.¹²⁴ die Verwendung von geschichteten Architekturen (vgl. Kap. 3.3), da einzelne Schichten unabhängig voneinander entwickelt und getestet werden können.

¹²¹ Erläuterungen der Prinzipien gemäß Vogel et al. (2009): Software-Architektur, S. 125 ff

¹²² Vogel et al. (2009): Software-Architektur, S. 134

¹²³ vgl. auch Starke (2009): Effektive Software-Architekturen, S. 153

Die genannten Architekturmittel wirken sich in signifikanter Weise auf die *Modularität* des Systems aus, die die wichtigsten der bisher genannten Aspekte der nichtfunktionalen Anforderungen zusammenfasst. Eine hohe Modularität liegt vor, wenn die Architektur aus wohldefinierten Bausteinen besteht, deren funktionale Verantwortlichkeiten klar abgegrenzt sind¹²⁵. Im Kontext des virtuellen Testens sowie der o.g. Maßnahmen zur Reduzierung der Systemkomplexität hilft die Modularisierung ebenfalls dabei, die Abhängigkeiten des Systems nach außen zu verringern, wenn beispielsweise Umfeldrepräsentations- und Interpretationsschnittstellen möglichst unabhängig vom verwendeten Sensorkonzept zur Verfügung stehen.

Auch die mit dem Themengebiet des Testens in Verbindung stehenden Entwicklungstendenzen der Absicherung und Freigabe gemäß ISO 26262 profitieren von den genannten Architekturprinzipien. Im dort vorgeschriebenen Prozessschritt der „Gefährdungsanalyse und Risikoabschätzung“ werden *funktionale Sicherheitsanforderungen* definiert, die dann hierarchisch, ausgehend von System- bis auf Komponentenebene, in *technische Sicherheitsanforderungen* (bzgl. Hard- und Software) herunter gebrochen werden^{126a}. Je klarer die Systemmodule in ihrer Zuständigkeit abgegrenzt sind, desto einfacher ist auch die Dekomposition der funktionalen Sicherheitsanforderungen in die einzelnen Module. So schreiben Wilhelm et al.^{126b}, dass „grundsätzlich großer Wert auf die strukturierende Systemarchitektur gelegt werden muss“. Dementsprechend wird in der Norm selbst beschrieben, dass eine modulare, hierarchisch aufgebaute Struktur gefordert wird, um die Systemkomplexität zu reduzieren und die Testbarkeit zu erhöhen¹²⁷. Ebenfalls bietet die ISO 26262 besondere Dekompositionsregeln an, sodass eine hohe funktionale Sicherheitsanforderung in weniger hohe technische Sicherheitsanforderungen mehrerer Module zerlegt werden kann (z. B. ASIL C in ASIL B und ASIL A)¹²⁸, sofern letztere eine ausreichende Unabhängigkeit voneinander besitzen. Daraus kann die Forderung abgeleitet werden, dass Systemfunktionen in möglichst voneinander unabhängige Teilfunktionen der Entscheidung und Ausführung aufgespalten werden. Hillenbrand¹²⁹, der die Anforderungen der ISO 26262 auf E/E-Architekturen¹³⁰ in Kraftfahrzeugen untersucht, bezeichnet eine solche Maßnahme als eine sog. *strukturelle Redundanz*, die auf der Übereinstimmung von Ergebnissen zur Verhinderung uner-

¹²⁴ Bass et al. (2013): Software Architecture in Practice, S. 168

¹²⁵ Vogel et al. (2009): Software-Architektur, S. 145

¹²⁶ Wilhelm et al. (2015): Funktionale Sicherheit und ISO 26262. a: -; b: S. 97

¹²⁷ ISO 26262, Teil 4, Kap. 7.4.2

¹²⁸ ASIL: Automotive Safety Integrity Level; Dekompositionsregeln gemäß ISO 26262, Teil 9, Kap. 5

¹²⁹ Hillenbrand (2012): Funktionale Sicherheit nach ISO 26262..., S. 182 ff

¹³⁰ E/E: Elektrik / Elektronik

wünschter Ausführungen von Systemfunktionen beruht. Hinsichtlich der Systemarchitektur empfiehlt Hillenbrand¹³¹ daher das Einfügen eines solchen Redundanzmittels, das in der Regel das Einfügen einer zusätzlichen Hierarchieebene umfasst.

Änderbarkeit & Erweiterbarkeit

Funktionale Systemarchitekturen besitzen idealerweise eine lange Lebensdauer, da eine vollständige Umstellung oder grundlegende Überarbeitung einer Architektur sehr aufwändig und daher kostenintensiv ist. Dies steht im Gegensatz zur Hardware-Architektur, die sich im selben Zeitraum stark ändern kann, z. B. durch Einführung neuer Technologien¹³². Darunter zählen z. B. neue Datenbusse, Gateways, Umfeldsensoren oder auch neue Technologien wie V2X (Vehicle-To-X) oder Backendsysteme. Daher ist es wichtig, dass die funktionale Architektur möglichst unabhängig gegenüber diesen absehbaren Änderungen ist. Im Kontext des automatisierten Fahrens werden sich voraussichtlich auch die Rolle des Fahrers sowie dessen Einbindung in das Fahrzeugführungssystem grundlegend ändern. So schreibt Winner¹³³, dass in Analogie zur Entwicklung moderner Unterhaltungselektronik radikal geänderte Bedienkonzepte Schlüsselfaktoren in unmittelbarer Vergangenheit darstellten, was sich auch im automotiven Bereich wiederholen könne. Als Beispiel hierfür zeigt Franz¹³⁴ verschiedene Bedienelemente für eine manöverbasierte Fahrzeugführung, darunter ein Touchpad mit Gestenerkennung. Deshalb ist es für eine funktionale Architektur auch notwendig, gegenüber Änderungen im Bereich der Mensch-Maschine-Schnittstellen robust zu sein.

Abowd et al.¹³⁵ konkretisieren den Begriff der Änderbarkeit, indem sie diesen in vier Kategorien aufspalten:

- Die Erweiterung / Verbesserung der Systemfunktion
- Das Löschen unerwünschter Systemfunktionen
- Die Adaption an neue Umgebungs- / Betriebsbedingungen
- Das Umstrukturieren der Architektur (z. B. aufgrund geänderter nichtfunktionaler Anforderungen)

Für die vorliegende Arbeit ist vor allem der Punkt der Adaption an neue Umgebungsbedingungen relevant, da dieser die o.g. vorhersehbaren Änderungen der Hardware und MMS beinhaltet. Von hoher Bedeutung im Bereich der Architekturen für Fahrerassis-

¹³¹ Hillenbrand (2012): Funktionale Sicherheit nach ISO 26262..., S. 186

¹³² vgl. Maurer (2013): Automotive Engineering: A Personal Perspective, S. 25

¹³³ Winner (2015): Quo Vadis, FAS?

¹³⁴ Franz (2014): Entwicklung und Evaluation eines Interaktionskonzepts...

¹³⁵ Abowd et al. (2007): SAAM: A Method for Analyzing the Properties of Software Architectures, S. 7

tenzsysteme ist üblicherweise auch die Erweiterbarkeit der Systemfunktionen. Diese Anforderung ist jedoch bereits durch die funktionalen Anforderungen gegeben, in denen gefordert wird, dass das zu entwickelnde System assistiertes sowie teil- und vollautomatisiertes Fahren in den Domänen Autobahn, Land- und Stadtstraße umfassen soll (vgl. Kap. 2.2.3). Auf eine darüber hinausgehende Erweiterbarkeit, z. B. die Anwendung des Systems in unstrukturiertem Gelände (z. B. Off-Road-Fahren) wird verzichtet. Das Löschen von Systemfunktionen sowie das Umstrukturieren der Architektur wird im Weiteren ebenfalls nicht betrachtet, da dies Maßnahmen sind, die sich aus (im Vergleich zu dieser Arbeit) geänderten funktionalen und nichtfunktionalen Anforderungen ergeben.

Bass et al.^{136a} beschreiben, dass die Werkzeuge zur Erhöhung der Änderbarkeit nahezu identisch mit den im vorherigen Abschnitt erläuterten Maßnahmen zur Erhöhung der Testbarkeit sind, d.h. ebenfalls die Systemkomplexität durch die Architekturprinzipien der hohen Kohäsion, losen Kopplung und Separation of Concerns zu verringern. Bass et al.^{136b} schlagen zusätzlich vier spezifische Entwicklungstaktiken vor:

- **Reduktion der Modulgrößen:** Besitzt ein Modul eine hohe Anzahl an funktionalen Verantwortlichkeiten, so ist der Aufwand größer, dieses Modul zu ändern.
- **Erhöhen der (semantischen) Kohäsion:** Dienen mehrere Verantwortlichkeiten in einem Modul nicht demselben Zweck, so sollten diese in mehrere Module ausgelagert werden.
- **Verringerung der Modul-Kopplung,** u.a. durch die *Refaktorisierung*. Dabei sollen „unschöne“ Teile¹³⁷ der Architektur und des Quelltextes aufgespürt und beseitigt werden, z. B. doppelte (redundante) Zuständigkeiten und deren Quelltext-Passagen, die dann in ein separates Modul ausgelagert werden.
- **Verzögerte Festlegung (von Parametern):** Quelltext-spezifische Maßnahme, bei der (z. B. mathematische) Funktionen möglichst allgemein gehalten werden, sodass spätere Funktionsänderungen oder Erweiterungen schnell umzusetzen sind.

2.3.3 Zusammenfassung

Um relevante nichtfunktionale Anforderungen für den Architekturentwurf des A/A-Systems in der vorliegenden Arbeit definieren zu können, wurden allgemeine Qualitäts-

¹³⁶ Bass et al. (2013): Software Architecture in Practice. a: S. 167; b: S. 121 ff

¹³⁷ Hierbei wird auch von sog. „Code-Smell“, also „übelriechendem Quelltext“ gesprochen, die faktisch einem schlechten Architektur- bzw. Programmierstil gleichzusetzen sind. Eine ausführliche Übersicht dieser „Smells“ ist verfügbar unter http://de.wikipedia.org/wiki/Smell_%28Programmierung%29, letzter Zugriff am 03.10.16

attribute für Softwareprodukte (vgl. Kap. 2.3.1) den Fachdomänen-spezifischen Herausforderungen des automotiven Entwicklungsumfelds gegenübergestellt. Von herausragender Bedeutung für die Architekturentwicklung wurden zum einen die Testbarkeit und zum anderen die Änderbarkeit bzw. Erweiterbarkeit identifiziert. Für beide Anforderungskategorien wurden entsprechend des Standes der Technik spezifische Entwicklungstaktiken beschrieben, die die Erreichung der nichtfunktionalen Anforderungen unterstützen. Bei Betrachtung dieser Taktiken fällt auf, dass für beide Anforderungskategorien das Architekturprinzip der Modularität von großer Bedeutung ist und sich die anzuwendenden Taktiken in diesem Punkt überschneiden. Eine Übersicht über die nichtfunktionalen Anforderungen sowie deren Entwicklungstaktiken ist in Abbildung 2-1 gegeben.

Im weiteren Verlauf der Arbeit werden Architekturvarianten erarbeitet und hinsichtlich der Erfüllung der nichtfunktionalen Anforderungen verglichen. Um diesen Arbeitsschritt zu erleichtern, wird auf Basis von Abbildung 2-1 eine Checkliste zur Überprüfung der nichtfunktionalen Anforderungen erstellt, die eine qualitative Bewertung und Vergleich der Varianten ermöglichen soll. Das Ergebnis dieses Arbeitsschrittes ist in Tabelle 2-2 dargestellt, die die nichtfunktionalen Anforderungen des vorherigen Abschnitts zusammenfasst. Der Vorteil der Checkliste ist es, detailliertere Bewertungskriterien als beispielsweise die reine Forderung nach Modularität heranziehen zu können.

An dieser Stelle sei darauf hingewiesen, dass die Checkliste keine quantitativen oder Quellcode-spezifischen Bewertungsverfahren erlaubt bzw. beinhaltet. Solche Maßnahmen benötigen ein bereits implementiertes Gesamtsystem und sind somit für die Entwicklung der funktionalen Systemarchitektur ungeeignet.

Im folgenden Kapitel wird gemäß der vorgestellten Entwicklungsmethodik (vgl. Abbildung 1-5) der Stand der Forschung und Technik von Basis- und Referenzarchitekturen aus den Domänen der A/A-Systeme und Robotik vorgestellt, die als Basis zur Definition eines eigenen Architekturentwurfs dienen.

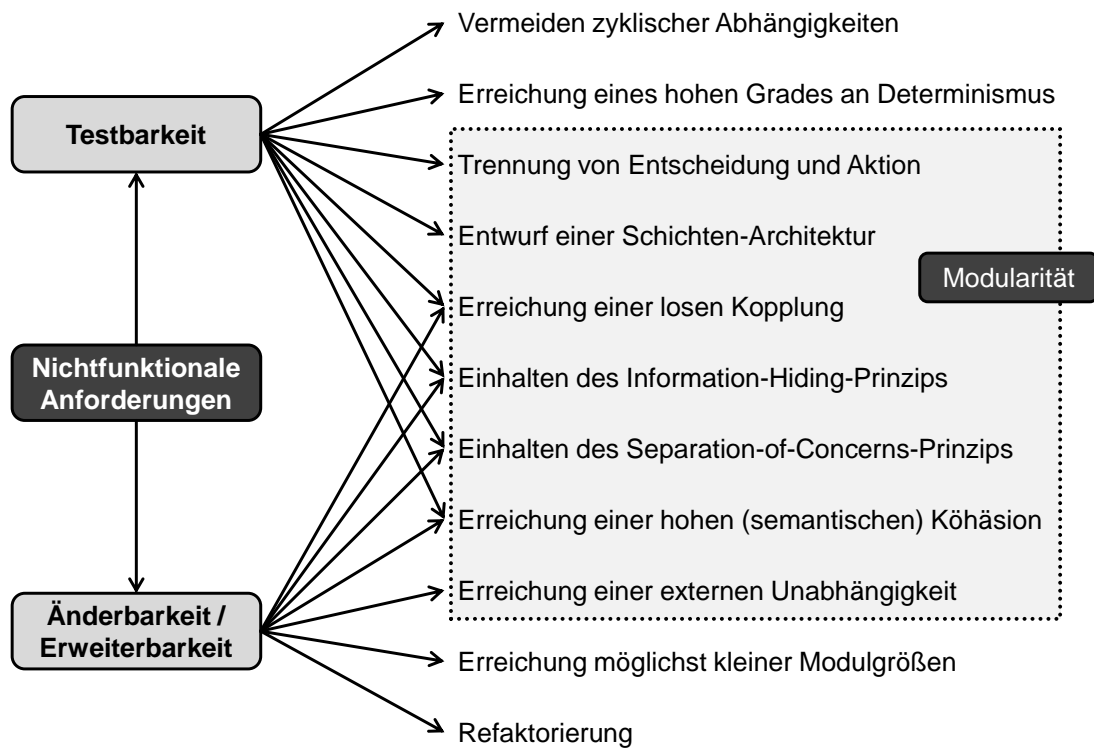


Abbildung 2-1: Übersicht der nichtfunktionalen Anforderungen und deren Merkmale¹³⁸

¹³⁸ Eigene Darstellung unter Verwendung der gesammelten Anforderungen aus Abschn. 2.3.2

Tabelle 2-2: Checkliste zur Prüfung der nichtfunktionalen Anforderungen

Nr.	Kriterium	Beschreibung
1	Prüfung auf zyklische Abhängigkeiten (Kreisbezüge)	Analyse der Informations- und Kontrollflüsse, die zur Umsetzung der funktionalen Anforderungen entstehen. Prüfung, ob dabei möglichst wenige zyklische Abhängigkeiten auftreten.
2	Prüfung auf Determinismus	Prüfung, ob bei vorgegeben Modul-Eingangsgrößen das Ergebnis bzw. das resultierende Systemverhalten mit hoher Konfidenz vorhergesagt werden kann.
3	Prüfung auf Möglichkeit der Dekomposition	Prüfung, ob zwischen der Entscheidung und Ausführung von Systemfunktionen unterschieden wird und diese hinreichend unabhängig voneinander sind.
4	Prüfung auf hohe Kohäsion	Prüfung, ob ein Modul alle notwendigen Teilfunktionen, Eigenschaften und Kompetenzen besitzt, um eine spezifizierte Gesamtfunktion umsetzen zu können.
5	Prüfung auf lose Kopplung	Prüfung, ob eine Modulveränderung leicht möglich ist, ohne dass ein komplexes Verständnis anderer Module vorausgesetzt wird.
6	Prüfung auf Einhaltung des Information-Hiding-Prinzips	Prüfung, ob die Architektur mehrere (Abstraktions-) Schichten besitzt ¹³⁹ .
7	Prüfung auf Einhaltung des Information-Hiding-Prinzips	Prüfung, ob zur Umsetzung der funktionalen Anforderung nur Schnittstellen benutzt werden müssen, die eine (Hierarchie-) Ebene entfernt sind (nur bei Schichtenarchitekturen).
8	Prüfung auf Separation-of-Concerns	Prüfung, ob funktionale Anforderungen gemäß erforderlicher Teilfunktionen gruppiert und in ein einzelnes Modul abgebildet werden können.
9	Prüfung auf Separation-of-Concerns	Prüfung, ob technische Aspekte (Hardware, Hardware-Architektur) bei der Dekomposition keine signifikante Rolle spielen bzw. nicht ausschlaggebend für das Ergebnis sind.
10	Prüfung auf externe Unabhängigkeit	Prüfung, ob sich der Änderungsaufwand bei Änderungen in der Hardware-Architektur auf wenige Module beschränkt.
11	Prüfung der Modulgröße	Prüfung, ob sich die Zuständigkeiten eines Moduls nicht weiter in mehrere Module aufspalten lassen.
12	Prüfung auf mögliche Refaktorisierung	Prüfung, ob Zuständigkeiten nicht redundant vorliegen oder widersprüchlich sind.
13	Prüfung auf semantische Kohäsion	Prüfung, ob sich die Funktionalität eines Moduls semantisch einfach und eindeutig beschreiben lässt.

¹³⁹ Hierbei sei angemerkt, dass das Vorhandensein von Abstraktionsschichten die Einhaltung des Information-Hidings-Prinzips in der Regel begünstigt, jedoch alleine keine hinreichende Bedingung darstellt. Aus diesem Grund wird Eintrag Nr. 6 im folgenden Verlauf der Arbeit eher als Indiz gewertet.

3 Funktionale Architekturen für automatisierte Fahrzeuge

Bei der Analyse des Standes der Technik fällt auf, dass im Bereich klassischer Fahrerassistenzsysteme nur wenige Quellen zu finden sind, die das Thema der Entwicklung funktionaler Systemarchitekturen tiefgehend behandeln. Dies steht in einem beachtlichen Gegensatz zur Domäne der Robotik, in der zahlreiche Basis- und Referenzarchitekturen für automatisierte Fahrzeuge (nicht zwangsläufig auf Straßenfahrzeuge begrenzt) vorgestellt und vor allem zwischen den Befürwortern verschiedener Architektur-Paradigmen leidenschaftlich diskutiert werden¹⁴⁰.

Als Gliederung für das vorliegende Kapitel wird eine Einteilung von Basis-Architekturen aus dem Bereich der Robotik herangezogen. In der Literatur werden dabei oftmals übereinstimmend drei Gruppen aufgeführt: *Sequenzielle Architekturen*, *parallele (nebenläufige) Architekturen* sowie *hybride Architekturen*¹⁴¹. Diese werden im Folgenden näher beschrieben. Im Anschluss daran werden konkrete Anwendungsbeispiele dieser Basis-Architekturen gegeben, die sich aus Forschungsprojekten des Standes der Technik ableiten lassen.

3.1 Sequenzielle Basis-Architekturen

Sequenzielle Architekturen sind in der Literatur in der Regel als *Sense-Plan-Act* (SPA) oder auch *Sense-Model-Plan-Act* (SMPA) -Architekturen bekannt^{142 143}. In dieser Bezeichnung ist inhärent die grundlegende Funktionsbeschreibung enthalten: So werden die Signalverarbeitungsschritte der Sensordatenerfassung, der Umgebungsmodellierung, der Planung und schließlich der Aktionsausführung sequenziell hintereinander ausgeführt, wie in Abbildung 3-1 dargestellt ist. SPA-Architekturen wurden besonders in den Anfängen der Robotik genutzt, um z. B. einfache fahrende Roboter zu entwickeln¹⁴⁴.

¹⁴⁰ u.a. Albus (1997): The NIST Real-time Control System...

¹⁴¹ vgl. u.a. Hertzberg et al. (2012): Mobile Roboter, S. 317 ff

¹⁴² Hertzberg et al. (2012): Mobile Roboter, S. 321

¹⁴³ Kortenkamp & Simmons (2008): Robotic Systems Architectures and Programming, S. 189

¹⁴⁴ wie z. B. „Shakey the Robot“ in den 1960er Jahren, entwickelt von der Stanford Universität. Vgl. Kortenkamp & Simmons (2008): Robotic Systems Architectures and Programming, S. 189

Darüber hinaus wurde laut Aussage von Hertzberg et al.^{145a} diese Basis-Architektur nie in der gezeigten Reinform eingesetzt.

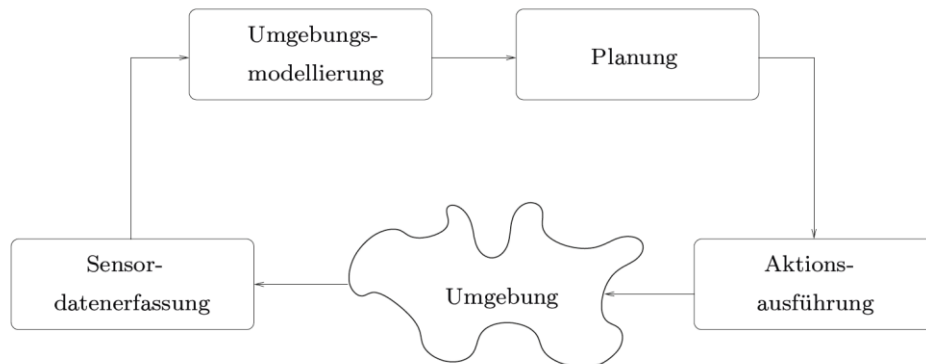


Abbildung 3-1: Abstrahierte Form einer SMPA-Architektur¹⁴⁶

Neben der in Abbildung 3-1 gezeigten, einfachen Form existieren in der Literatur weitere verwandte Varianten, die auch als *funktionsorientierte* oder *deliberative Architekturen* bezeichnet werden^{147a 148}. Der Begriff *funktionsorientiert* spiegelt dabei die klare Partitionierung der Systemfunktionen (Sensorsystem, Modellierung, Planung, Ausführung) auf die jeweiligen Systemmodule wieder und der Begriff *deliberativ*, dass mit jeder Aktionsausführung eine Planung verbunden ist, um das Missionsziel zu erreichen. Funktionsorientierte Architekturen können gemäß Hamdi^{147b} auch hierarchisch gegliederte Schichten besitzen, die z. B. die (Aktions-) Planung in mehrere sequentiell zu durchlaufende Schritte unterteilen (bspw. Routenplanung und Bahnplanung).

Allen Varianten der sequenziellen Architekturen ist gemein, dass die Informationsverarbeitungsschritte namensgemäß sequenziell durchlaufen werden, was jedoch insbesondere in einer sich ständig ändernden bzw. unvorhersehbaren Einsatzumgebung zu großen Problemen führt. Der Grund dafür ist, dass die (Bewegungs-) Planung des Systems sowie die Aktions-Ausführung auf einem *Weltmodell* beruht, das bei dessen Erstellung aufgrund der rechen- und zeitintensiven Prozessschritte bereits veraltet sein kann^{145b}. Dieser Umstand bremst somit den Kontrollzyklus des Systems und erhöht dessen Reaktionszeit. Hamdi^{147c} kommt zu der Erkenntnis, dass aus diesem Grund der erreichbare Grad an Autonomie eines solch strukturierten Systems stark eingeschränkt ist, da die Forderung nach ausreichender System-Reaktivität und gleichzeitig nach einer hohen

¹⁴⁵ Hertzberg et al. (2012): Mobile Roboter. a: S. 321; b: S. 322

¹⁴⁶ entnommen aus Hertzberg et al. (2012): Mobile Roboter, S. 321

¹⁴⁷ Hamdi (1999): Entwurf adaptiver lernender Roboter. a: S. 33 ff; b: S. 34-35; c: S. 37

¹⁴⁸ Pellkofer (2003): Verhaltensentscheidung für autonome Fahrzeuge..., S. 41 ff

Detaillierung des Weltmodells - unabhängig von der weiteren Entwicklung der Rechenleistung von Computern bzw. Steuergeräten - in Zielkonflikt zueinander stehen.

3.2 Parallele Basis-Architekturen

Parallele Basis-Architekturen begegnen dem Problem mangelnder Reaktivität durch konsequente Parallelisierung der Informationsverarbeitungsschritte. Ein anschauliches Beispiel dafür ist in Abbildung 3-2 gegeben. Die Funktionen der Umgebungsmodellierung, Planung und Aktionsauswahl sowie der Aktionsausführung laufen dabei parallel ab, sodass die Gesamtrechnzeit für einen Programmzyklus im Vergleich zum seriellen Ansatz nicht mehr durch die Summe aller Laufzeiten der Einzelmodule, sondern durch den Zyklus der „langsamsten“ Funktion gegeben ist ^{149a}.

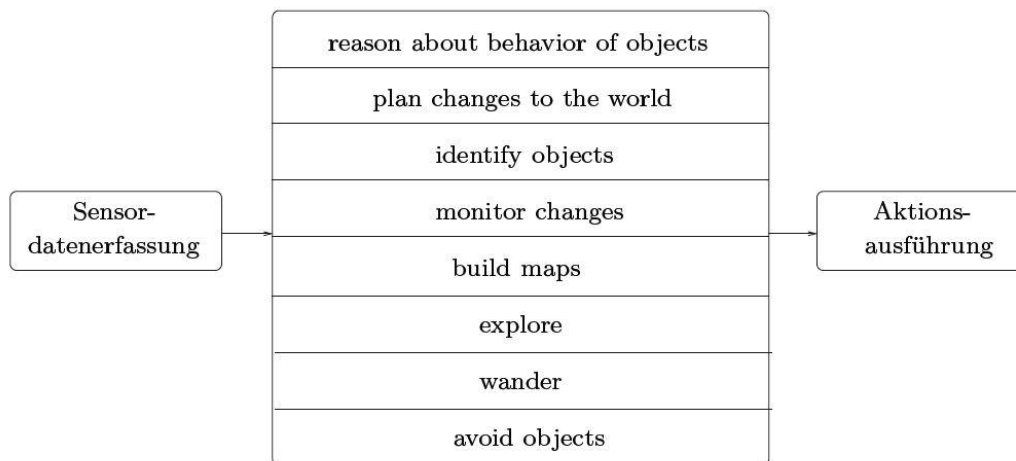


Abbildung 3-2: Abstrahierte Form einer parallelen Basis-Architektur¹⁵⁰

Die in Abbildung 3-2 gezeigten Funktionen werden in der Literatur als Verhaltenselemente, Verhaltensbausteine oder schlicht als *Verhalten* bezeichnet, weshalb sich als Synonym für parallele Basis-Architekturen auch der Begriff der *verhaltensbasierten Architekturen* etabliert hat. Die Grundidee verhaltensbasierter Architekturen ist es, dass die einzelnen Verhaltenselemente in jedem Zyklus durch direkten Zugriff auf die Sensordaten ihren Beitrag zur Kontrolle des Roboters leisten und so das effektivste Kontrollkommando bestimmen ^{149b}. Die Verhaltenselemente entsprechen dabei typischen Teilfunktionen, die zur Erreichung des gewünschten Ziels notwendig sind.

¹⁴⁹ Hertzberg et al. (2012): Mobile Roboter. a: S. 320; b: S. 322

¹⁵⁰ entnommen aus Hertzberg et al. (2012): Mobile Roboter, S. 323

In der Reinform verhaltensbasierter Systeme entspricht ein *reaktives* Verhalten einer direkten Reaktion auf ein Eingangssignal¹⁵¹. Im regelungstechnischen Sinne stellt dies eine festgelegte Übertragungsfunktion dar^{152a}. Dabei besitzen reaktive Verhalten kein „Gedächtnis“ und verzichten gänzlich auf eine (symbolische) Abstraktion der Umwelt^{153 152b}. In der Literatur werden reaktive Verhalten deshalb oftmals mit (aus der Biologie bekannten) Reflexen verglichen^{152b}. Um komplexere Aufgaben lösen zu können, müssen mehrere Verhalten miteinander kombiniert werden. Die Art und Weise, wie dies erfolgt, unterscheidet verschiedene Unterkategorien verhaltensbasierter Architekturen.

Einer der ersten und prominentesten Befürworter verhaltensbasierter Architekturen war Rodney A. Brooks mit der sog. *Subsumption-Architecture*¹⁵⁴. Der Begriff „Subsumption“ steht dabei frei übersetzt für die Unterordnung von Verhalten gegenüber anderen Verhalten¹⁵⁵ oder analog das Überstimmen eines Verhaltens durch ein anderes Verhalten¹⁵³. Die einzelnen Verhalten werden dabei als Zustandsautomaten realisiert, die miteinander über mehrere Ebenen verknüpft sind, wie in Abbildung 3-3 gezeigt wird.

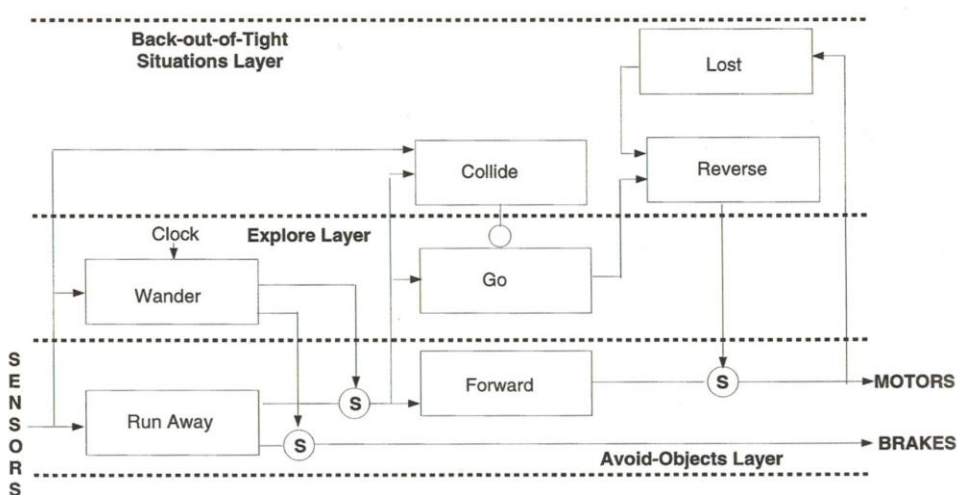


Abbildung 3-3: Subsumption-Architecture nach Brooks für ein einfaches Fahrzeug¹⁵⁶

¹⁵¹ „A behavior, simply put, is a reaction to a stimulus“; Aus: Arkin (2000): Behavior-Based Robotics, S. 66

¹⁵² Arkin (2000): Behavior-Based Robotics. a: S. 80; b: S. 66

¹⁵³ Hertzberg et al. (2012): Mobile Roboter, S. 323

¹⁵⁴ siehe z. B. Brooks (1986): A Robust Layered Control System for a Mobile Robot

¹⁵⁵ Das Subsumieren bedeutet das Unterordnen bzw. Unterstellen, s. Duden Online <http://www.duden.de/rechtschreibung/subsumieren>, letzter Zugriff am 03.10.16

¹⁵⁶ entnommen aus Arkin (2000): Behavior-Based Robotics, S. 134

Kern der Subsumption-Architecture ist dabei die Möglichkeit, dass Verhalten einer höheren (Kompetenz-) Ebene die Ausgabe von Verhalten einer untergeordneten Ebene unterdrücken können, was in Abbildung 3-3 durch das Symbol „S“ für „Surpress“ symbolisiert wird. Im gezeigten Beispiel kann entsprechend das Verhalten „Wander“ (freies Fahren) das Verhalten „Run Away“ (von einem Hindernis Wegdrehen) unterdrücken, wenn kein Hindernis vorhanden ist, und im umgekehrten Fall das Verhalten „Forward“ (vorwärts Fahren) unterdrücken und „Run Away“ aufschalten, wenn ein Hindernis vorhanden ist. Arkin^{157a} charakterisiert eine solche Verhaltenskoordination als *konkurrierend* mit einer fest programmierten Hierarchie bzw. Priorisierung.

Arkin selbst schlägt stattdessen in seinem eigenen Ansatz, der *Motor-Schema-Architektur*, eine *kooperative* Verhaltenskoordination vor, die durch Beobachtungen aus der Biologie inspiriert ist^{157b}. Dabei werden mehrere sog. Motor-Schemata erzeugt, die unterschiedliche Ziele verfolgen (z. B. Vorwärtsbewegen, Zielerreichen, Hindernisausweichen etc.). Jedes Schema erzeugt dabei ein Potenzialfeld, aus dem sich ein resultierender Bewegungsvektor im zwei- oder dreidimensionalen Raum berechnen lässt. Mit Hilfe einer einfachen Vektoraddition wird dann das resultierende Gesamtsystemverhalten berechnet.

Dem inhärenten Problem paralleler Architekturen, dass mehrere Verhalten häufig gleichzeitig um die Nutzung der Aktoren des Fahrzeugs konkurrieren, setzen die Entwickler mehr oder weniger komplexe *Arbitrierungsmechanismen* entgegen. Diese werden in der Literatur auch als *Command-Fusion* bezeichnet¹⁵⁸. Neben den beiden bereits gezeigten Methoden führt Arkin^{157c} noch zwei weitere Möglichkeiten auf:

- *Action-Selection (konkurrierend)*: Dabei wird das Verhalten übernommen, das über den höchsten *Aktivierungs-Level* verfügt, der vom Verhalten selbst situationsabhängig berechnet wird. Im übertragenen Sinne fallen in diese Kategorie auch *Maximum-* oder *Minimum-Select-Mechanismen*.
- *Wahl-Prozess (kooperativ)*: Dem Verhalten stehen dabei eine endliche Anzahl an Entscheidungsoptionen zur Auswahl. Jedes Verhalten stimmt dann für die aus seiner Sicht beste Option. Am Ende gewinnt die Entscheidungsoption mit den meisten Stimmen.

Ein Beispiel für eine parallele Architektur mit Wahl-Prozess ist die von Rosenblatt¹⁵⁹ entwickelte *Distributed Architecture for Mobile Navigation* (DAMN). Dabei stimmen verschiedene Verhalten für eine gewünschte zukünftige Bahnkrümmung des automatisierten Fahrzeugs, wobei die Stimmen der einzelnen Verhalten dabei zusätzlich von

¹⁵⁷ Arkin (2000): Behavior-Based Robotics. a: S. 133; b: S. 141 ff; c: S. 111 ff

¹⁵⁸ z. B. Sukthankar (1997): Situation Awareness for Tactical Driving, S. 108 ff

¹⁵⁹ Rosenblatt (1997): DAMN: a distributed architecture for mobile navigation

einem Arbitrierungsmodul situationsspezifisch gewichtet werden. Die Krümmung mit den meisten Stimmen gewinnt schließlich und wird aktiv geschaltet, siehe Abbildung 3-4. Die Fahrzeuggeschwindigkeit wird im Anschluss an die resultierende Krümmung angepasst, sodass ein sicheres Befahren der Bahn möglich ist.

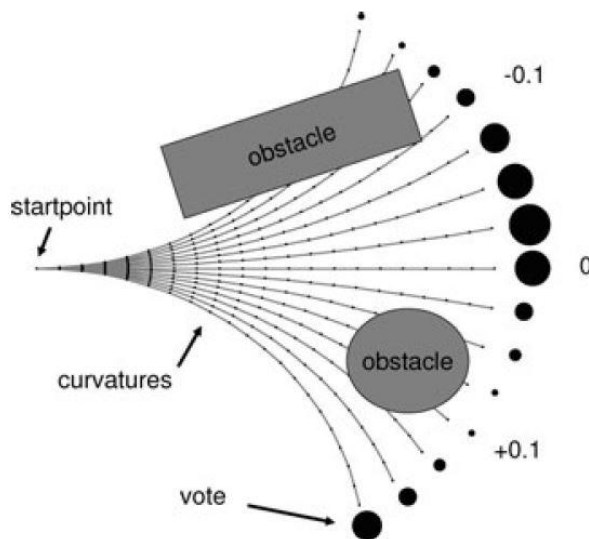


Abbildung 3-4: Anwendungsbeispiel für die DAMN-Architektur¹⁶⁰

Unabhängig von der Lösung des Arbitrierungsproblems gilt die parallele, verhaltensbasierte Roboterarchitektur aufgrund der hohen Verbreitung als sehr erfolgreich¹⁶¹. Durch die Reaktivität können die Systeme in einer sich stark dynamisch ändernden Umwelt eingesetzt werden. Darüber hinaus gelten sie aufgrund der Gliederung in Verhaltens-elemente als gut modularisierbar und vergleichsweise einfach zu implementieren¹⁶².

Die Leistungsfähigkeit des Gesamtsystems steigt dabei durch das Hinzufügen von zusätzlichen Verhaltenselementen stetig an, sodass ein außenstehender Beobachter den Eindruck erhalten kann, dass der Roboter über die Fähigkeit vernünftigen und logischen Denkens verfügt¹⁶³. Hamdi spricht in diesem Zusammenhang von einem *quasi-intelligent* erscheinendem Verhalten¹⁶³, das sich rein durch das Zusammenspiel primitiver Module, ohne Nutzung eines verhaltensinternen Zustandsspeichers oder Modellannahmen der Umgebung ergibt. Diese Besonderheit ist auch unter dem Begriff der *Emergenz* bekannt, d.h. dass die Gesamtfähigkeiten eines Systems größer sind als die Summe

¹⁶⁰ entnommen aus Rauskolb et al. (2008): Caroline: An Autonomously Driving Vehicle..., S. 701

¹⁶¹ u.a. Kortenkamp & Simmons (2008): Robotic Systems Architectures and Programming, S. 190

¹⁶² vgl. u.a. Hamdi (1999): Entwurf adaptiver lernender Roboter, S. 41-42

¹⁶³ Hamdi (1999): Entwurf adaptiver lernender Roboter, S. 38

der Einzelfähigkeiten¹⁶⁴. Bei Prüfung der Fachliteratur entsteht der Eindruck, dass gerade parallele, verhaltensbasierte Architekturen eine besonders hohe Emergenz besitzen, was von den Entwicklern, wie z. B. Brooks, auch gezielt propagiert wurde¹⁶⁵.

Dennoch scheint die „Effizienz“ verhaltensbasierter Architekturen bei komplexer werdenden Problemen stark abzunehmen. Dies begründet sich dadurch, dass dafür mehr und mehr Verhalten implementiert und im Anschluss arbitriert werden müssen. Kortenkamp & Simmons¹⁶⁶ schreiben dazu:

“However, behavior-based robots soon reached limits in their capabilities. It proved very difficult to compose behaviors to achieve long-range goals and it proved almost impossible to optimize robot behavior.”

In diesem Zitat wird ebenfalls der Umstand angesprochen, dass eine Verfolgung von spezifischen Missionszielen nicht direkt möglich ist, da keine deliberativen Elemente existieren, die eine längerfristige Missionsplanung ermöglichen. Als Beispiel hierfür kann der heute erhältliche Staubsaugerroboter herangezogen werden, der sehr häufig aus Kostengründen über eine Zufallsbewegung, den sog. *random walk*, verfügt¹⁶⁷. Eine solche Funktion lässt sich prinzipiell sehr gut mit der Subsumption-Architecture implementieren (vgl. Abbildung 3-3). Ein zielgerichtetes Abfahren eines Raumes ist mit einer klassischen verhaltensbasierten Architektur nicht möglich, da dafür eine Kartierung des Raumes, ein Pfad-Planungsmodul sowie eine Lokalisierungsfunktion notwendig wären¹⁶⁷ - also Verhalten, die sich nicht über eine einfache Übertragungsfunktion ohne interne Speichervariablen implementieren lassen.

Diese Problematik führte schließlich zur Entwicklung von hybriden Basis-Architekturen¹⁶⁶, die im nächsten Abschnitt vorgestellt werden.

3.3 Hybride Basis-Architekturen

Hybride Architekturen kamen zu ihrer Bezeichnung, da sie sowohl über deliberative als auch über reaktive Systemelemente verfügen und somit die vorgestellten parallelen und sequenziellen Architekturkonzepte verbinden. Dabei sollen die Vorteile beider Ansätze bei gleichzeitiger Minimierung der Nachteile kombiniert werden¹⁶⁸. Die deliberativen

¹⁶⁴ Arkin spricht auch von emergentem Verhalten, vgl. Arkin (2000): Behavior-Based Robotics, S. 105

¹⁶⁵ vgl. Arkin (2000): Behavior-Based Robotics, S. 131

¹⁶⁶ Kortenkamp & Simmons (2008): Robotic Systems Architectures and Programming, S. 190

¹⁶⁷ Kim et al. (2007): Human-Aided Cleaning Algorithm for Low-Cost Robot Architecture, S. 367 ff

¹⁶⁸ Hamdi (1999): Entwurf adaptiver lernender Roboter, S. 44

Elemente dienen dabei der Verfolgung langfristiger Ziele, z. B. der Erfüllung der Fahrmission. Reaktive Elemente sind für die situativ richtige und zeitadäquate Umsetzung der Sensorsignale in Aktorbefehle zuständig¹⁶⁹. Hybride Architekturen erlauben die Rekonfiguration reaktiver Systemanteile auf Basis eines Weltmodells und der Fähigkeit, Schlussfolgerungen bzgl. der unterlagerten Verhaltenselemente zu ziehen^{170a}. Diesbezüglich stellt sich die Frage, wie hybride Basisarchitekturen funktional aufgebaut werden, also wie die deliberativen und reaktiven Elemente auf Modulebene partitioniert sind und zusammenwirken. Arkin^{170b} schreibt hierzu:

„The central issue in differentiating the many approaches to hybrid architectures [...] focuses on interface design: What is the appropriate boundary for the subdivision of functionality? How is coordination effectively carried out? This is one of the most interesting and pressing research areas in intelligent robotics today.”

Obwohl eine konkrete Antwort darauf noch aussteht und stark abhängig von der Anwendungsdomäne ist, scheint sich die Fachwelt einig darin zu sein, dass insbesondere *hierarchisch strukturierte Architekturen* mit mehreren *Ebenen* bzw. *Schichten* eine adäquate Grundstruktur dafür bereitstellen^{170c}. In diesem Zusammenhang wird auch von *hierarchischen Mehrebenensystemen* gesprochen¹⁷¹. Gemäß Arkin^{170c} sind dabei mindestens zwei Ebenen notwendig, nämlich je eine zur Repräsentation der reaktiven und deliberativen Anteile. Sehr häufig finden sich jedoch Architekturen mit genau drei Ebenen.

Warum Systemarchitekten unabhängig voneinander scheinbar immer wieder zu diesem Ergebnis kommen, ist unbekannt. Da jedoch die Vermittlung bzw. die Koordination zwischen den deliberativen und reaktiven Anteilen eine komplexe Aufgabe darstellt, bietet sich laut Arkin^{170c} eine dritte Ebene an, die diese Koordinationsaufgabe leistet. Dabei wird die mittlere Ebene häufig als ein Aktionsauswahlmodul realisiert, das komplexe Handlungsfolgen zur Zielerreichung in eine weniger komplexe Sequenz von Steuerungsparametern zerlegt, die sich auf die (reaktiven) Verhaltensbausteine abbilden lassen^{169 172}. Eine entsprechende typische architektonische Struktur wird in Abbildung 3-5 gezeigt. Die darin enthaltenen Pfeile, die zwischen den Ebenen nicht nur hierarchisch von oben nach unten laufen, sondern in beide Richtungen zeigen, deuten an, dass untere Ebenen den höheren Ebenen Rückmeldung über ihren derzeitigen Ausführungszustand geben, sodass sich Fehler oder unvorhergesehene Situationen auch dahingehend auswirken können, dass ggf. ein neuer Plan erstellt werden muss.

¹⁶⁹ Hamdi (1999): Entwurf adaptiver lernender Roboter, S. 44

¹⁷⁰ Arkin (2000): Behavior-Based Robotics. a: S. 207; b: S. 212; c: S. 213

¹⁷¹ vgl. Maurer (2000): Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen, S. 31 ff

¹⁷² vgl. Hertzberg et al. (2012): Mobile Roboter, S. 324

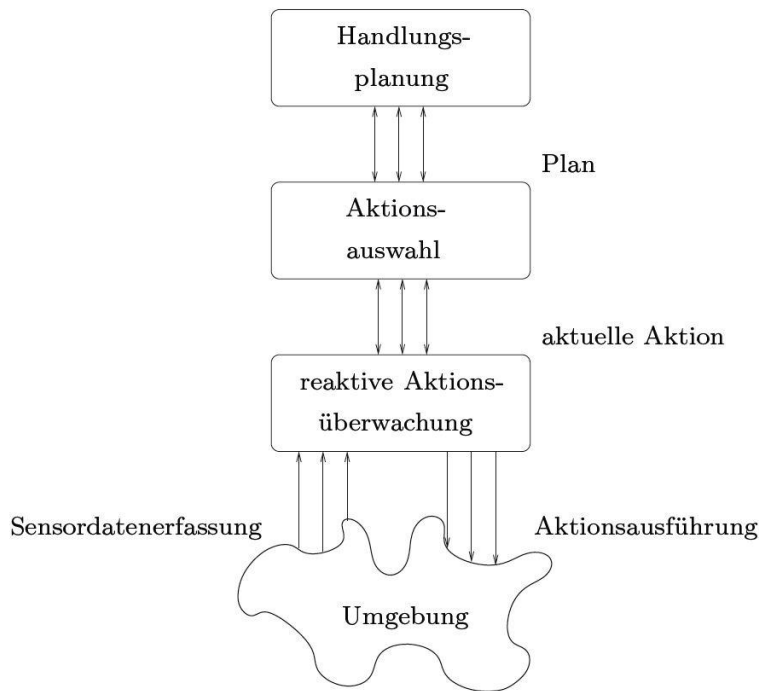


Abbildung 3-5: Abstrahierte Form einer hybriden Basis-Architektur¹⁷³

Unabhängig von der Anzahl der Ebenen besitzen hierarchische Mehrebenensysteme die Gemeinsamkeit, dass höhere Hierarchieebenen einen höheren *Abstraktionsgrad* hinsichtlich der Kompetenzen zur Problembeschreibung und -Lösung besitzen als niedrigere. Maurer¹⁷⁴ konkretisiert dies wie folgt:

- Je höher eine Einheit in der Hierarchie angesiedelt ist, umso größer ist der Ausschnitt vom Gesamtsystem, den die Einheit überblickt.
- Die Entscheidungsperiode auf einer höheren Ebene ist länger als die auf der darunterliegenden.
- Eine Einheit auf einer höheren Ebene ist mit langsameren Aspekten des Systemverhaltens befasst.
- Beschreibungen und Probleme auf höheren Ebenen sind weniger gut strukturiert, mit mehr Unsicherheit behaftet und schwieriger quantitativ zu formalisieren.

Dies entspricht auch der Beschreibung von Arkin¹⁷⁵, wobei dieser ergänzt, dass höhere Ebenen in der Regel einen größeren räumlichen Bereich (*spatial scope*) in dessen Umwelt überblicken als untere Ebenen.

¹⁷³ entnommen aus Hertzberg et al. (2012): Mobile Roboter, S. 325

¹⁷⁴ wörtliches Zitat nach Maurer (2000): Flexible Automatisierung von Straßenfahrzeugen mit Rechnern, S. 34

¹⁷⁵ Arkin (2000): Behavior-Based Robotics, S. 210

In der Fachliteratur wird eine Vielzahl von hybriden Basis-Architekturen beschrieben, die im Rahmen der Arbeit aus Umfangsgründen nicht alle aufgeführt werden können. Eine der bekanntesten Beispiele, die auf die soeben beschriebene Abstraktionshierarchie eingeht, ist jedoch die Arbeit von Albus¹⁷⁶ mit der *Real-Time Control System Architecture* (RCS), die vom US National Institute of Standards and Technology als eine Weiterentwicklung der *NASA Standard Reference Model Architecture* (NASREM)¹⁷⁷ herausgegeben wurde. Die RCS-Architektur ist ein hierarchisches Mehrebenensystem, wobei jede Ebene über ein Weltmodell verfügt, das zur Abstraktionsebene passende Karten und Zustandsvariablen der Systemumgebung erstellt sowie Ereignisse verfolgt, wie in Abbildung 3-6 zu sehen ist. Ebenfalls existiert in jeder Ebene ein Modul zur Verhaltensgenerierung, das Pläne auf Basis der Zielvorgaben einer höheren Systemebene ableitet und diese auf Basis des Weltmodells hinsichtlich einer Kosten- / Nutzenbetrachtung bewerten lässt (*value judgment*). Erfolgversprechende Pläne werden dann an eine untergeordnete Ebene weitergeleitet, die über die gleiche Struktur verfügt. Der Abstraktionsgrad ändert sich für jede Ebene hinsichtlich der räumlichen und zeitlichen Auflösung ca. um den Faktor 10, sodass z. B. eine Ebene, die für die Missionserreichung zuständig ist, einen zeitlichen Horizont von mehreren Stunden besitzt und die niedrigste, Akteur-nahe Ebene im Bereich von Millisekunden arbeitet.

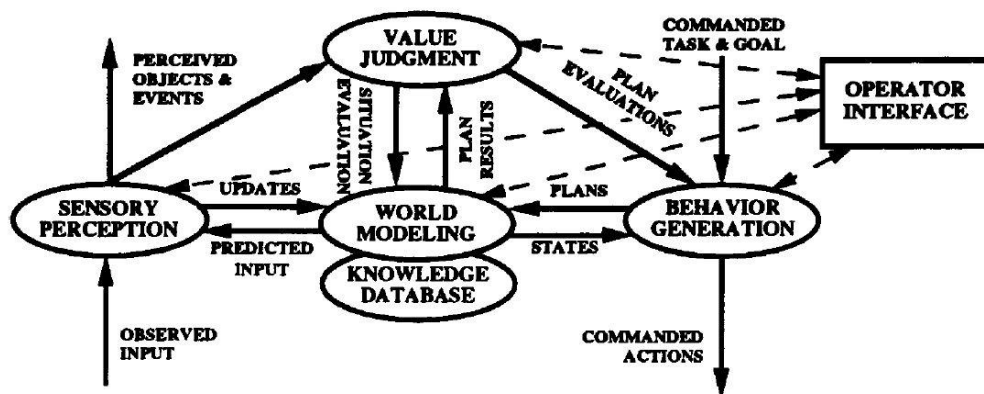


Abbildung 3-6: Ausschnitt aus der RCS-Architektur von Albus¹⁷⁸

Die RCS-Architektur schreibt dabei nicht vor, wie viele Ebenen erforderlich sind und wie diese genau funktional zu partitionieren sind. Dies deutet laut Maurer¹⁷⁹ an, dass es

¹⁷⁶ Albus et al. (1997): The NIST Real Time Control System (RCS)...

¹⁷⁷ Albus et al (1994): NASREM - The NASA/NBS Standard Reference Model...

¹⁷⁸ entnommen aus Albus et al. (1997): The NIST Real Time Control System (RCS)...

¹⁷⁹ Maurer (2000): Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen, S. 38

sich bei der RCS-Architektur eher um ein Denkmodell handelt, das den Grundgedanken hybrider Architekturen in einer konsistenten Fachsprache zusammenfasst¹⁸⁰.

Dass hybride Architekturen eine geeignete Strukturierung für Systeme darstellen, die komplexe, zielgerichtete Aufgaben in einer sich dynamisch schnell ändernden Umwelt lösen müssen, scheint allgemeine Zustimmung in der Fachwelt zu finden, wie folgende Zitate belegen:

*„Es herrscht weitgehend Einigkeit, dass hybride Architekturen eine adäquate Struktur für Roboterkontrollsoftware darstellen [...]“*¹⁸¹

*„In der Zwischenzeit sind die meisten Entwickler von Robotersystemen davon überzeugt, dass für die Ausschöpfung des ganzen Potenzials verhaltensbasierter Ansätze sowohl deliberative als auch reaktive Komponenten notwendig sind.“*¹⁸²

*„One outcome of a 1995 workshop on robot architectures was the observation that a multi-layered hybrid architecture [...] is emerging as the architectural design of choice.“*¹⁸³

Dennoch gibt Pellkofer¹⁸² zu bedenken, „dass hybride Architekturen Kompromisse an beiden Enden des Spektrums eingehen“ und das Zusammenspiel zwischen deliberativen und reaktiven Komponenten noch nicht ausreichend verstanden zu sein scheint. Gerade in diesem Punkt ist deswegen weiterer Forschungsbedarf gegeben, der in dieser Arbeit aufgegriffen wird.

3.4 Anwendungsbeispiele für automatisierte Fahrzeuge

In diesem Abschnitt des Kapitels werden konkrete Anwendungsbeispiele der gezeigten Basis-Architekturen für den Bereich der automatisierten Fahrzeuge vorgestellt, die auch eine Auswahl der wichtigsten Forschungsprojekte und -gruppen aufzeigen sollen. Dabei kann jedoch kein Anspruch auf Vollständigkeit gegeben werden, u.a. da nicht in allen Projekten auf die funktionale Architektur eingegangen wird. Eine weiterführende Übersicht über Konzepte des automatisierten Fahrens ist u.a. in Matthaei et al.¹⁸⁴ sowie in

¹⁸⁰ vgl. auch Kortenkamp & Simmons (2008): Robotic Systems Architectures and Programming, S. 192

¹⁸¹ Hertzberg et al. (2012): Mobile Roboter, S. 325 - 326

¹⁸² Pellkofer (2003): Verhaltensentscheidung für autonome Fahrzeuge mit Blickrichtungssteuerung, S. 43

¹⁸³ Arkin (2000): Behavior-Based Robotics, S. 213

¹⁸⁴ Matthaei et al. (2015): Autonomes Fahren

Matthaei¹⁸⁵ (insbesondere zur DARPA Urban Challenge) gegeben. Die Beispiele sind - soweit möglich - chronologisch sortiert.

3.4.1 Ansätze nach Dickmanns und Maurer

Ein Meilenstein in Bezug auf das automatisierte Fahren wurde mit dem sog. *4-D-Ansatz*¹⁸⁶ für dynamisches maschinelles Sehen von Dickmanns an der Universität der Bundeswehr in München (UniBwM) erreicht. Der Ansatz basiert auf dem 3-Ebenen Modell für zielgerichtetes, menschliches Handeln nach Rasmussen (vgl. Kap. 5.1.1). Entsprechend beschreibt Dickmanns eine hierarchisch geschichtete Architektur mit drei Ebenen: Dem wissensbasierten und regelbasierten Verhalten sowie mit reflexartig implementierten Verhalten, wie z. B. *Lane-Keeping* und *Convoy-Driving*, auf der untersten Ebene^{187a}. Für den Fall, dass die Verhalten zur Fortführung der Mission nicht ausreichen, z. B. aufgrund eines Hindernisses, so agiert die mittlere Ebene als eine regelbasierte Vorsteuerung, die dann einen Fahrstreifenwechsel veranlassen kann. Entsprechend kann diese Art der Architektur als eine hybride Architektur angesehen werden.

Der 4-D-Ansatz wurde im Rahmen des Prometheus-Projektes¹⁸⁸, das zwischen 1987 und 1995 stattfand, von der UniBwM in die Versuchsfahrzeuge VaMoRs-L¹⁸⁹ (oder auch nur VaMoRs¹⁹⁰), ein Nutzfahrzeug, und später in VaMoRs-P^{187b} (oder auch kurz VaMP), ein Pkw, integriert¹⁹¹. Der Erfolg dieser Architektur führte dazu, dass Dickmanns und Albus (RCS-Architektur, siehe Abschn. 3.3) ihre beiden Ansätze in ein gemeinsames Denkmodell, der sog. *4-D/RCS Architektur*, abbildeten¹⁹². Maurer^{193a} resümiert dazu jedoch, dass die Kombination der beiden Ansätze in konzeptuellen Nachteilen resultiert.

Maurer^{193b} entwickelte stattdessen den 4-D-Ansatz weiter und erreichte eine flexible Automatisierung hinsichtlich des Funktionsumfanges des Gesamtsystems. Dabei wurde das verhaltensbasierte Konzept konkretisiert und um eine Überwachung der Leistungs-

¹⁸⁵ Matthaei (2015): Wahrnehmungsgestützte Lokalisierung in fahrstreifengenauen Karten..., S. 23 ff

¹⁸⁶ 4-D steht für den 3-dimensionalen Raum plus Zeit als zusätzliche Dimension

¹⁸⁷ Dickmanns et al. (1994): The Seeing Passenger Car „VaMoRs-P“. a: S. 70; b: -

¹⁸⁸ Programme for a European Traffic of Highest Efficiency and Unprecedented Safety

¹⁸⁹ siehe Behringer & Müller (1998): Autonomous Road Vehicle Guidance...

¹⁹⁰ VaMoRs: Versuchsfahrzeug für autonome Mobilität und Rechnersehen

¹⁹¹ vgl. auch Maurer (2000): Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen, S. 6 ff

¹⁹² Albus (2000): 4-D/RCS Reference Model Architecture for Unmanned Ground Vehicles

¹⁹³ Maurer (2000): Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen. a: S. 38; b: -

fähigkeiten der einzelnen Verhalten erweitert. Basierend auf dieser Leistungsfähigkeit, die über regelungstechnische Kriterien wie z. B. das Einschwingverhalten oder Totzeiten im Regelkreis ermittelt werden, kann der Grad der Automatisierung des Gesamtsystems zur Laufzeit auf den vom Fahrer geforderten Funktionsumfang angepasst werden. Kern des Ansatzes von Maurer ist ein Verhaltensentscheidungsmodul, das als hierarchischer Zustandsautomat implementiert wurde.

Als ein beeindruckendes Zeugnis der Leistungsfähigkeit der Ansätze von Dickmanns und Maurer gilt die über 1.600 km lange Fahrt von VaMP von München nach Odense in Dänemark im Jahr 1995, die zu ca. 95 % der Strecke automatisiert zurückgelegt werden konnte¹⁹⁴.

3.4.2 Ansätze im Rahmen der DARPA Urban Challenge

Die DARPA Grand Challenge (2004 und 2005) sowie die nachfolgende DARPA Urban Challenge (2007) erfuhren eine starke mediale Aufmerksamkeit. Dabei handelte es sich um einen von einer Behörde des US-Verteidigungsministeriums veranstalteten Wettbewerb, an dem im Falle der Urban Challenge insgesamt 89 Forschungsteams teilnahmen. Ziel war es, nachzuweisen, dass automatisierte Fahrzeuge einen Parcours innerhalb eines Vorstadt-Szenarios einschließlich anderen, von Menschen gesteuerten Fahrzeugen, bewältigen können¹⁹⁵.

Das Siegerfahrzeug „BOSS“ des Teams „Tartan Racing“ besitzt architektonisch eine hierarchisch aufgebaute Architektur mit drei Ebenen, der Missionsebene, einer Verhaltens-Exekutiv-Ebene sowie der Bewegungsplanungsebene¹⁹⁶. Die Missionsebene ist für die Routenplanung zuständig, die Verhaltens-Exekutiv-Ebene gliedert die Mission in Sequenzen von Teilzielen auf und die Bewegungsplanungsebene setzt diese Teilziele in Form von berechneten Fahrzeugtrajektorien um. Dass es sich bei der untersten Ebene um reaktive, verhaltensähnliche Elemente handelt, kann aus der Aussage abgeleitet werden, dass die potenziell möglichen Trajektorien „sehr schnell“ bzgl. ihrer Entfernung zu statischen und dynamischen Objekten bewertet werden. Entsprechend kann hier, trotz der geringfügigen Informationslage der Literatur, von der Nutzung einer hybriden Basis-Architektur ausgegangen werden.

Als bestes nicht-amerikanisches Team nahm das „Team Carolo“ der TU Braunschweig mit dem Fahrzeug „Caroline“ an der Urban Challenge teil. Die funktionale Architektur

¹⁹⁴ Maurer (2000): Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen, S. 135

¹⁹⁵ vgl. Ibanez-Guzman et al (2012): Autonomous Driving: Context and State-of-the-Art

¹⁹⁶ die Begriffe wurde aus dem Englischen übersetzt: Mission Planning, Behavioral Executive und Motion Planning; vgl. Baker et al. (2008): Robust Mission Execution for Autonomous Urban Driving.

beruht hierbei auf dem DAMN-Ansatz, der bereits in Abschn. 3.2 vorgestellt wurde. Entsprechende parallel laufende Verhalten sind laut Rauskolb et al.¹⁹⁷ beispielsweise „den Wegpunkten folgen“ (gemäß vorgegebener Route), „fahre im Fahrstreifen“ und „vermeide Hindernisse“¹⁹⁸. Jedes der Verhalten wählt dabei für jeden Punkt der zukünftigen Trajektorie eine Krümmung (siehe Abbildung 3-4). Dabei werden die Stimmen der Verhalten gemäß situativer Gesichtspunkte gewichtet. Um auch komplexere Situationen mit dem DAMN-Ansatz darstellen zu können, wurde eine „Interrupt“-Funktion eingefügt, um die Längsführung des Fahrzeugs, z. B. an Kreuzungen, unterbrechen zu können. Ob es sich bei der vom Team Carolo eingesetzten Architekturvariante noch um eine rein parallel angeordnete verhaltensbasierte Architektur handelt, ist aufgrund der Erweiterung durch die Interrupt-Funktion fraglich. Dennoch überwiegt hier die parallele Anordnung, da auch der deliberative Anteil der Funktion als eines unter mehreren Verhalten („Wegpunkten folgen“) implementiert wurde.

Neben der eigentlichen Zielsetzung der Veranstaltung, dem Bewältigen des Parcours, profitierte die Disziplin des automatisierten Fahrens maßgeblich dadurch, dass einige teilnehmende Teams ihre Projekte auf Basis der gemachten Erfahrungen weiterentwickelten, darunter z. B. Google Cars, die durch Beteiligung prominenter DARPA-Entwickler wie Sebastian Thrun oder Chris Urmson eine Art „Spin-Off“ darstellen, ebenso wie das Projekt „Stadtpilot“ der TU Braunschweig.

3.4.3 Stadtpilot

Im Projekt Stadtpilot gelang es der TU Braunschweig im Jahr 2010 mit dem Forschungsfahrzeug „Leonie“ autonom auf öffentlichen, innerstädtischen Straßen zu fahren. Dabei verfügt das Fahrzeug über die Fähigkeiten einem Fahrstreifen zu folgen, die Längsführung entsprechend der Verkehrsgegebenheiten anzupassen sowie an einer Lichtsignalanlagen-geregelten Kreuzung zu wenden¹⁹⁹. Mittlerweile werden auch Fahrstreifenwechsel autonom vorgeschlagen und (nach Bestätigung durch einen Sicherheitsfahrer) selbstständig ausgeführt²⁰⁰.

Der ursprüngliche aus der DARPA-Challenge bekannte und auf DAMN basierende Architekturansatz wurde für das Projekt Stadtpilot verworfen. Zwar attestierte man dem vorherigen Ansatz eine gewisse „Kreativität“, die jedoch aber zu nicht immer vorher-

¹⁹⁷ Rauskolb et al. (2008): Caroline: An Autonomously Driving Vehicle..., S. 700 ff

¹⁹⁸ Die Begriffe wurden frei aus dem Englischen übersetzt

¹⁹⁹ Nothdurft (2011): Stadtpilot: First Fully Autonomous Test Drives in Urban Traffic

²⁰⁰ vgl. Ulbrich & Maurer (2013): Probabilistic Online POMDP Decision Making for Lane Changes...

sagbarem Verhalten führte²⁰¹. Dieses nicht-deterministische Verhalten wurde in Bezug auf die gut strukturierte Umgebung des Braunschweiger Stadtrings als negativ bewertet, sodass ein regelbasierter Ansatz eingesetzt wurde, der ein höheres Maß an Vorhersagbarkeit und Sicherheit bietet. Aus diesem Grund wurde für Leonie nach Aussage von Wille^{202a} ein hybrides Entscheidersystem gewählt, das sich in zwei Ebenen, die Aktionsplanung und der Ausführung, unterteilen lässt. Die Aktionsplanung generiert dabei auf Basis der Fahrmission passende Fahrkorridore, die dem Trajektorienplaner auf Ausführungsebene als Optimierungsspielraum vorgegeben werden. Reaktive Elemente, wie z. B. das Fahrzeugfolgeverhalten, werden dabei an dieser Hierarchie vorbei direkt an die Längsregelung geben^{202b}.

In einer späteren Veröffentlichung von Ulbrich & Maurer²⁰³ wird im Zusammenhang von Stadtpilot von einer hierarchisch gegliederten Architektur gesprochen, die sich in die Navigationsebene (Routenplanung), die Bahnführungsebene (Manöverplanung), die Stabilisierungsebene (Trajektorienplanung) sowie die Manöverausführungs-Ebene (Hardware Ebene) untergliedert.

3.4.4 Conduct-by-Wire

Als letztes Beispiel wird das von der TU Darmstadt entwickelte Fahrzeugführungskonzept Conduct-by-Wire erläutert, auf dessen Basis der Modus *kooperative Automation* des PRORETA 3-Projektes aufbaut. Ziel von CbW ist es, manöverbasiertes Fahren zu ermöglichen, bei dem der Fahrer von der Aufgabe der Fahrzeugstabilisierung befreit ist und stattdessen mit dem Fahrzeug über zeitdiskrete Manövereingaben kommuniziert²⁰⁴. CbW wurde bisher nicht in einem realen Fahrzeug, sondern in einer Computersimulation implementiert.

Die Systemarchitektur basiert dabei auf dem 3-Ebenen-Modell nach Donges, wobei die oberste (Navigations-) Ebene prinzipbedingt entfällt²⁰⁵, bzw. dem Fahrer vorbehalten bleibt. Auf der Bahnführungs- bzw. Manöver-Ebene findet die Manöverplanung statt, d.h. eine Vermittlung des Fahrerwunsches, wobei als Ergebnis sog. *Fahrfunktionen* an die unterlagerte Trajektorienplanungsfunktion weitergegeben werden, die das gewünschte Verhalten des Fahrzeuges in je eine voneinander unabhängige longitudinale

²⁰¹ Wille et al. (2011): Der Stadtpilot - Autonomes Fahren auf dem Braunschweiger Stadtring

²⁰² Wille (2012): Manöverübergreifende autonome Fahrzeugführung.... a: S. 66-67; b: S. 102

²⁰³ Ulbrich & Maurer (2013): Probabilistic Online POMDP Decision Making for Lane Changes...

²⁰⁴ vgl. Franz et al. (2015): Conduct-by-Wire

²⁰⁵ Geyer (2013): Maneuver-Based Vehicle Guidance Based on the Conduct-by-Wire Principle

und laterale Komponente untergliedern²⁰⁶. Durch Kombination dieser Fahrfunktionen lassen sich auch komplexere Verhalten erzeugen, wie z. B. das Anpassen der Fahrgeschwindigkeit an ein Fahrzeug auf einem benachbarten Fahrstreifen, um ein Fahrstreifenwechsel vorzubereiten. Geyer²⁰⁷ bezeichnet die Architektur auch als einen kaskadierten Regelkreis. Aus diesem Grund kann auch hier von einem hybriden Architekturansatz gesprochen werden.

3.5 Zusammenfassung

Im Rahmen des vorliegenden Kapitels wurden aus der Disziplin der Robotik bekannte Basis-Architekturen vorgestellt. In der Literatur wird dabei ausschließlich und übereinstimmend von drei Varianten gesprochen, der sequenziellen, parallelen und hybriden Basisarchitekturen, auch wenn zuweilen alternative Begriffe gewählt werden (deliberativ, reaktiv & kombiniert). Darüber hinaus wurden in Abschn. 3.4 ausgewählte Forschungsprojekte als Anwendungsbeispiele aufgeführt. Gemäß der Definition aus Kap. 1.2.3 handelt es sich dabei um Referenzarchitekturen, da diese im weiteren Sinne eine architektonische Gesamtlösung für ein konkretes Anwendungsziel unter Berücksichtigung der projektspezifischen funktionalen Anforderungen darstellen. Auch dabei kann festgestellt werden, dass sich die gefundenen Referenzarchitekturen hinsichtlich ihrer Grundstruktur in die drei genannten Basis-Architekturen eingliedern lassen.

Aufgrund der Erkenntnisse dieses Kapitels erfolgt im weiteren Verlauf der Arbeit eine Auswahl der Basis-Architektur als erste Stufe des sequenziellen heuristischen Problemlösungsprozesses (vgl. Kap. 1.5).

²⁰⁶ vgl. Hakuli et al. (2011): Integriertes Konzept für die manöverbasierte Fahrerassistenz

²⁰⁷ Geyer (2013): Maneuver-Based Vehicle Guidance Based on the Conduct-by-Wire Principle, S. 114

4 Auswahl der Basis-Architektur

Nach der Definition der funktionalen sowie nichtfunktionalen Anforderungen in Kap. 2 sowie einer Recherche zu Basis- und Referenzarchitekturen aus dem Stand der Technik in Kap. 3 kann nun mit dem dritten Architekturentwicklungsschritt gemäß Abbildung 1-5 begonnen werden, nämlich der Festlegung der Architektur-Grobstruktur für das in dieser Arbeit vorliegende Problem.

Gemäß dem heuristischen Gesamtentwicklungsprozess werden im Folgenden Architekturvarianten ermittelt, die den Lösungsraum möglichst vollständig abdecken. Jede Variante wird dann zunächst hinsichtlich der Eignung zur Erfüllung der funktionalen Anforderungen bewertet. Falls eine der Varianten diese nicht erfüllt, entspricht das einer Falsifikation des Ansatzes und wird folglich ausgeschlossen. Varianten, die grundsätzlich in der Lage sind, die funktionalen Anforderungen zu erfüllen, werden anschließend hinsichtlich der Erfüllung der nichtfunktionalen Anforderungen (Qualitätskriterien) bewertet. Hierbei kommt die in Kap. 2.3.3 entwickelte Checkliste (Tabelle 2-2) zum Einsatz.

4.1 Varianten der Basis-Architektur

Als Basis für die Variantenentwicklung wird die aus dem vorherigen Kapitel bekannte Gliederung in sequenzielle, parallele und hybride Architekturen genutzt. Zunächst wird versucht, die funktionalen Anforderungen aus Kap. 2.2.3 auf die Basis-Architekturen abzubilden. Gelingt dies nicht, so scheidet die Variante aus dem Architekturentwicklungsprozess aus.

4.1.1 Sequenzielle Variante

Gemäß Kap. 3.1 zeichnet sich die sequenzielle Basisarchitektur dadurch aus, dass alle notwendigen Signalverarbeitungsschritte nacheinander ablaufen und sämtliche Aktionen des Fahrzeugs auf einem übergeordneten Plan zur Erreichung des spezifischen Missionsziels basieren. Abbildung 4-1 zeigt eine mögliche Grobstruktur der sequenziellen Architekturvariante, wenn die funktionalen Anforderungen dieser Arbeit in Form eines strukturiertes Gedankenexperiments darauf abgebildet werden würden. Dabei sei darauf hingewiesen, dass die aufgeführten Unterpunkte nicht zwangsläufig als einzelne Funktionsmodule implementiert sein müssen.

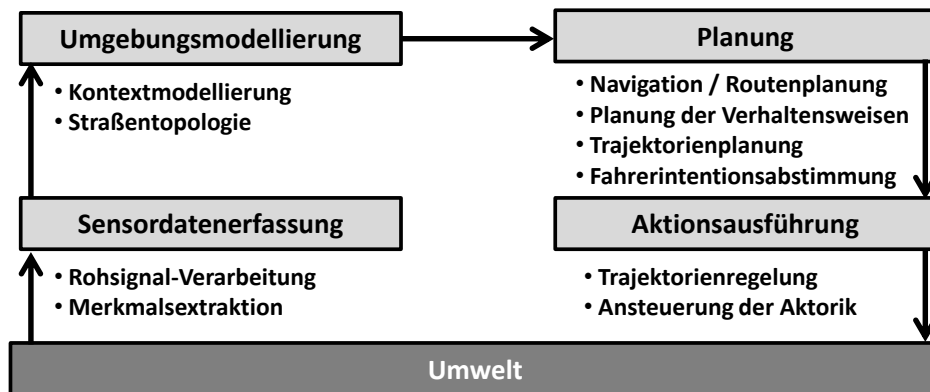


Abbildung 4-1: Grobstruktur der sequenziellen Architekturvariante

Im ersten Schritt erfolgt die Sensordatenerfassung, die sich gemäß Darms²⁰⁸ in die Signalverarbeitung und Merkmalsextraktion gliedern lässt. Eine ggf. notwendige Sensordatenfusion kann ebenfalls diesem Funktionsmodul zugeordnet werden. Im zweiten Schritt findet eine Umgebungsmodellierung statt, wobei als Ergebnis eine geeignete Repräsentation und Interpretation der Fahrzeugumgebung stehen muss, die hinreichend für die nachfolgende (Aktions-) Planung des automatisierten Fahrzeugführungssystems ist. Im Kontext der Anforderungen dieser Arbeit müssen demnach nicht nur quasi-kontinuierliche, geometrische Informationen²⁰⁹ über relevante statische und dynamische Objekte, wie z. B. die relative Lage und Bewegung zu anderen Fahrzeugen oder Fahrbahnbegrenzungen, vorhanden sein. Ebenfalls wird eine Kontextmodellierung benötigt, die die Verkehrsobjekte mit den Verkehrsstrukturelementen (z. B. Fahrstreifen) sowie deren Intentionen und den vorherrschenden Verkehrsregeln in einen semantischen Kontext zueinander bringt²¹⁰. Darüber hinaus muss eine Information über die Straßentopologie vorliegen, um im vollautomatisierten Modus im dritten Schritt des Kontrollzyklus eine Routenplanung zu ermöglichen. Auf Basis dieses Plans sowie der aktuellen Verkehrssituation muss eine konkrete Vorgabe bzw. Auswahl der auszuführenden Verhaltensweise (vgl. Tabelle 2-1) einschließlich der dazugehörigen Trajektorienplanung erfolgen. Insgesamt müssen gemäß den funktionalen Anforderungen der vorliegenden Arbeit im Planungsschritt zusätzlich die Wünsche des Fahrers hinsichtlich des Automationsgrades sowie ggf. der gewünschten Fahrmanöver berücksichtigt werden. Dazu ist es vorstellbar, dass bspw. die Planung der Route und / oder der auszuwählenden Verhaltensweisen entfällt, sofern der Fahrer nicht vollautomatisiert, sondern manöverbasiert /

²⁰⁸ Darms (2007): Eine Basis-Systemarchitektur zur Sensordatenfusion von Umfoldsensoren ..., S. 9

²⁰⁹ vgl. Matthaei (2015): Wahrnehmungsgestützte Lokalisierung in fahrstreifengenauen Karten..., S. 34

²¹⁰ Die Bedeutung dieser Informationen wird insbesondere bei Verhaltensweisen mit Interaktionen, wie z. B. beim Einfädelnlassen eines anderen Fahrzeugs auf den eigenen Fahrstreifen oder beim Abbiegen an Kreuzungen ersichtlich (vgl. Übersicht der Verhaltensweisen in Tabelle 2-1)

assistenten fahren möchte. Im letzten Schritt wird die geplante Aktion ausgeführt, die schließlich in der Ansteuerung der Aktorik mündet.

Bevor nun im weiteren Vorgehen die sequenzielle Architekturvariante hinsichtlich der Erfüllung der nichtfunktionalen Anforderungen überprüft wird, muss näher diskutiert werden, ob die in Abbildung 4-1 gezeigte Struktur auch tatsächlich die funktionalen Anforderungen zur Laufzeit erfüllen könnte. Eine notwendige Anforderung dafür ist, dass die Zykluszeit des gesamten Kontrollzyklus zu jeder Zeit zur notwendigen Regel-frequenz der zeitkritischsten Funktion des Systems passt. Da mit dem Zielsystem auch Brems- und Lenkeingriffe zur Unfallvermeidung umgesetzt werden müssen, die hohe dynamischen Anforderungen an den Regelkreis stellen, ergibt sich eine zulässige Zykluszeit im niedrigen Millisekunden-Bereich²¹¹. Unabhängig von der Art der Implementierung und der eingesetzten Rechner-Hardware ist eine Erfüllung dieser Anforderung fraglich, da die Summe der Laufzeiten *aller* beteiligter Algorithmen ausschlaggebend ist, was folglich bedeutet, dass jedes Modul eine Laufzeit im einstelligen Millisekunden-Bereich nicht überschreiten dürfte. Unvorteilhaft ist dies insbesondere hinsichtlich der Laufzeiten von Modulen, die weniger hohe dynamische Anforderungen an den Regelkreis stellen, wie beispielsweise die Kontextmodellierung und die darauf aufbauende Verhaltensauswahl. Matthaei²¹² legt hier bspw. Zykluszeiten von 100 - 500 ms zugrunde. Als Folge davon sinkt in unnötiger Weise die mögliche erreichbare Qualität der Umfeldmodellierung. Eine sequenzielle Basisarchitektur hätte weiterhin zur Folge, dass die Kontrollzykluszeiten unvorhersehbar schwanken können, beispielsweise wenn eine Routenneuplanung erfolgt. In einem solchen Fall würde sich die Phasenverschiebung im Regelkreis erhöhen, was zu Stabilitätsproblemen führen könnte.

Auf Basis dieser Erkenntnisse, insbesondere hinsichtlich der gravierenden Einschränkungen der Modul-Laufzeiten sowie der negativen Auswirkungen auf die Regelkreisstabilität erscheint eine Erreichung der dieser Arbeit zugrundeliegenden funktionalen Anforderungen mit Hilfe einer ausschließlich sequenziellen Architektur nicht möglich. Zu einer ähnlichen Einschätzung kommt auch Hamdi²¹³, der schreibt:

„Der Versuch, die Aktionen eines solchen Programms intelligenter zu machen, kann zu unzufrieden stellenden Ergebnissen führen. Je mehr Zeit das Programm der Auflösung von Konflikten zwischen den Sensordaten, der Verfeinerung des Weltmodells und der Optimierung des Plans widmet, desto länger wird die Verzögerung zwischen

²¹¹ Kfz-Drehratensensoren als Istwert-Geber für Fahrzeugstabilisierungsfunktionen senden ca. alle 15 ms neue Daten, siehe z. B. http://rb-aa.bosch.com/boaasocs/index.jsp;jsessionid=4BAC1A458CB2D7BC3B2DA88F0035279C?ccat_id=138&prod_id=528, letzter Zugriff am 27.08.2016

²¹² Matthaei (2015): Wahrnehmungsgestützte Lokalisierung in fahrstreifengenaue Karten..., S. 36

²¹³ Hamdi (1999): Entwurf adaptiver lernender Roboter, S. 37

Sensoren und Aktoren. [...] Es ist eine Verschwendung von Leistung, anzuhalten und die ganze Umwelt zu katalogisieren, bevor irgendeine Funktion ausgeführt wird“.

Hertzberg et al.²¹⁴ resümieren analog:

„Die SMPA-Architektur hat einen Denkfehler fest eingebaut [...]: Sie berücksichtigt nicht die Tatsache, dass es nicht die eine einzige Kontrollzyklusfrequenz gibt, sondern [...] unterschiedliche, die zu Kontrolldaten auf unterschiedlichen Granularitätsebenen passen. Die SMPA-Architektur schlägt alles gleichermaßen über den einen Leisten der langsamsten Frequenz. Würde man sie nicht nur als Denkmodell, sondern als reale Roboterkontrollarchitektur einsetzen, wäre klar, dass das nicht gut funktioniert.“

Aus diesen Gründen wird diese Architekturvariante im weiteren Verlauf der Arbeit nicht weiter detailliert und scheidet aus dem Bewertungsverfahren aus.

4.1.2 Parallele Variante

Um die vorliegenden funktionalen Anforderungen in eine parallele Basisarchitektur zu übertragen, werden die Funktionselemente aus Abbildung 4-1 prinzipbedingt in eine parallele Anordnung gebracht, wie in Abbildung 4-2 zu sehen ist. Dabei lassen sich jedoch nicht alle der vier Haupt-Informationsverarbeitungsschritte parallelisieren, denn alle Verhalten benötigen eine vorgeschaltete Sensordatenerfassung und eine nachgeschaltete Aktionsausführung. Ebenfalls auffällig ist, dass die Funktionselemente zur Planung, d.h. die Routenplanung sowie die Planung zur Sequenzierung der Verhaltensweisen, entfallen, da verhaltensbasierte Architekturen bewusst auf diese Elemente verzichten. Stattdessen muss die Zielerreichung ebenfalls als eigenes Verhalten dargestellt werden, wobei denkbar ist, dass ein solches Modul das Ziel verfolgt, einer vorgegebenen Fahrtrichtung oder einer vorgegebenen Sequenz an Wegpunkten zu folgen, wie es im Beispiel des Fahrzeugs „Caroline“ mit der DAMN-Architektur der Fall ist (vgl. Kap. 3.4.2). Die in Kap 2.2.3 identifizierten Verhaltensweisen (vgl. Tabelle 2-1) sind ebenfalls vollständig in der parallelen Anordnung präsent.

²¹⁴ Hertzberg et al. (2012): Mobile Roboter, S. 32

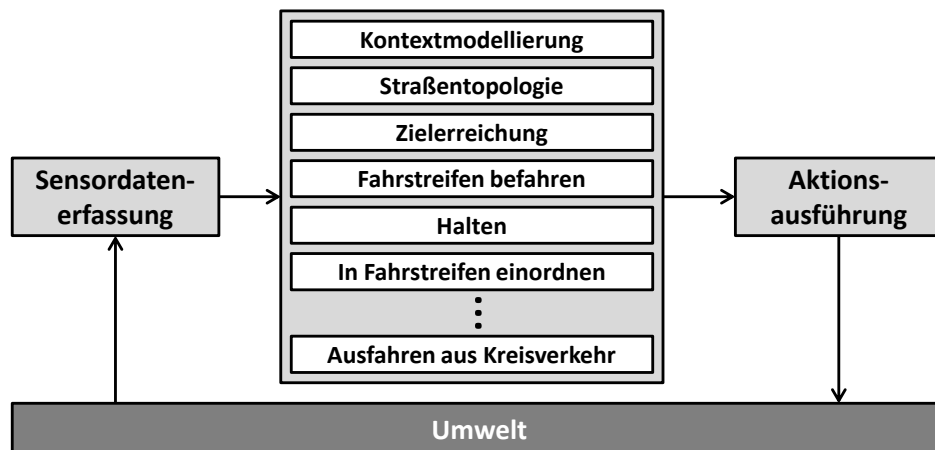


Abbildung 4-2: Grobstruktur der parallelen Architekturvariante

Alleinig auf Basis von Abbildung 4-2 ist jedoch nicht ersichtlich, wie die Architektur die funktionalen Anforderungen tatsächlich umsetzt. Wie in Kap. 3.2 erläutert wurde, spielt bei parallelen Basisarchitekturen die Art der Koordination der Verhaltensweisen eine übergeordnete Rolle, wobei in kooperative und konkurrierende Arbitrierungsmechanismen unterschieden werden kann. Aus diesem Grund werden im Folgenden die zwei Varianten getrennt betrachtet.

Eine geeignete Quelle dazu stellt Reichardt^{215a} dar, der beide Varianten in seiner Dissertation gegenüberstellt und diskutiert. Als Beispiel für die kooperative Verhaltenskoordination führt er, ähnlich wie bereits in Arkin²¹⁶ geschildert, einen generalisierten Potenzialfeldansatz als Paradigma für eine Gefahrenkarte auf und erweitert ihn, sodass auch Verkehrsregeln darin repräsentiert werden können^{215b}. So werden beispielsweise Fahrstreifen in Form einer „Wanne“ modelliert, wobei das Minimum der Karte in der Fahrstreifenmitte liegt, und andere Verkehrsteilnehmer als Erhebung repräsentiert. Die Idee der Gefahrenkarte überträgt Reichardt auch auf Verkehrsregeln, da eine Missachtung dieser indirekt zu einer Gefahr für das eigene Fahrzeug werden kann. So beschreibt er u.a. das Rechtsfahrgebot auf Autobahnen damit, dass das Potenzialfeld ein leichtes Gefälle nach rechts besitzt, oder das Rechtsüberholverbot durch eine räumliche Ausdehnung des Potenzialgebirges eines auf dem linken Fahrstreifen fahrenden Fahrzeugs auf den rechten Fahrstreifen. Mit Hilfe dieses Ansatzes ist es auch denkbar, dass Fahrmanöver, die der Fahrer im Falle einer Teilautomatisierung an das System delegieren kann, berücksichtigt werden. So könnte ein Fahrstreifenwechsel initiiert werden, indem ein Potenzialfeld erzeugt wird, das nach links oder rechts geneigt ist. Da jedes Verhalten ein eigenes Potenzialfeld erzeugt, können die resultierenden Teilkarten rechnerisch in eine gemeinsame, resultierende Karte fusioniert werden, wobei Reichardt eine

²¹⁵ Reichardt (1996): Kontinuierliche Verhaltenssteuerung eines autonomen Fahrzeugs.... a: -; b: S. 39 ff

²¹⁶ Arkin (2000): Behavior-Based Robotics, S. 98 ff

Maximumsbildung vorschlägt, in der die einzelnen Karten mit einem variablem Faktor normiert werden^{217a}. Übertragen auf die Aufgabenstellung dieser Arbeit könnten die Gewichte der in Tabelle 2-1 gezeigten Verhaltensweisen durch das Verhalten der „Zielerreichung“ so modifiziert werden, sodass sich die korrekten Richtungsentscheidungen, wie z. B. durch Abbiegen an Kreuzungen, mit einer hohen Wahrscheinlichkeit durchsetzen.

Ob dieses Gedankenmodell auch in der Praxis den funktionalen Anforderungen gerecht wird, diskutiert Reichardt^{217b} ausführlich und führt u.a. folgende Bedenken auf:

- Das Fahrverhalten ist ohne mindestens eine weitere logische Entscheidungskomponente nur schwer parametrierbar. Als Beispiel führt Reichardt einen Überholvorgang auf, wobei das langsamer fahrende Fahrzeug einerseits passiert werden soll, es jedoch gleichzeitig eine imaginäre abstoßende Kraftkomponente auf das eigene Fahrzeug ausübt, was paradoxerweise zu einer Abbremsung führt.
- Die Korrektheit der Reaktion des Fahrzeugs ist nur schwer zu gewährleisten, da zum einen die Potenzialwerte der einzelnen Teilkarten und zum anderen die Gewichtungsfaktoren Einfluss auf das Gesamtergebnis besitzen. Die ursprünglich modellierten Verhalten werden nicht in der intendierten (Rein-) Form umgesetzt.
- Sog. schematische Handlungen, wie z. B. das Setzen des Fahrtrichtungsanzeigers vor einem Fahrstreifenwechsel, werden nicht abgedeckt.

Weiterhin ist zu ergänzen, dass sich im Falle des manöverbasierten Fahrens das vom Fahrer gewünschte Manöver nicht zwangsläufig durchsetzen muss. Somit wäre die geforderte Manöverdelegation eher als Manöverempfehlung zu bezeichnen.

Aufgrund der genannten Nachteile untersucht Reichardt^{217c} ebenfalls eine Umsetzung mittels einer konkurrierenden Verhaltenskoordination in Form eines sog. Inhibitionsnetzwerks, das eine große Ähnlichkeit zur Subsumptions-Architektur von Brooks besitzt (vgl. Kap. 3.2). Auf Basis der Fahrsituation werden dabei eine variable Menge an Verhaltensmustern instanziiert²¹⁸, die sich in die Hauptgruppen „Notreaktionen“, „Basisverhaltensmuster“ (z. B. Abstandhalten, Spurhalten) und „schematisierte Handlungen“ (Manöver und Reaktion auf Verkehrsregeln) unterteilen lassen. Verhaltensmuster besitzen eine feste Basis-Gewichtung und die Möglichkeit, sich untereinander zu hemmen, woraus sich ein situationsspezifisches, variables Gewicht berechnen lässt. Über eine Berechnungsvorschrift des Inhibitionsnetzwerkes werden letztlich gemäß dem variablen Gewicht die Steuervorgaben für die Längs- und Querbeschleunigung des

²¹⁷ Reichardt (1996): Kontinuierliche Verhaltenssteuerung eines autonomen Fahrzeugs.... a: S. 48; b: S. 49 ff; c: S. 57 ff

²¹⁸ variabel deshalb, da z. B. für jedes Verkehrsobjekt ein Basisverhaltensmuster „Abstand halten“ instanziiert wird

Fahrzeugs anteilig verrechnet und somit fusioniert. Der Hauptunterschied zur oben geschilderten kooperativen Verhaltenskoordination liegt darin, dass Verhaltensmuster, die durch Hemmung anderer Verhaltensmuster ein negatives Gewicht erhalten, vollständig deaktiviert werden und somit keinen Einfluss mehr auf das Gesamtergebnis ausüben können.

Die beiden verhaltensbasierten Architekturvarianten wurden im Kontext der knotenpunktfreien Autobahnfahrt eingesetzt und dienen somit nach eigener Aussage Reichardts „einem reaktiven Fahrzeugverhalten in einem beschränkten Einsatzgebiet“²¹⁹. Um den Ansatz auf weitere Anwendungsfälle, wie z. B. Landstraßen und Stadtverkehr, erweitern zu können, schlägt Reichardt das Hinzufügen einer Situationsanalyse mit höherem Abstraktionsgrad ("wissensbasierte" Planung mit höherer Vorrauschaureichweite, die die Handlungsabsichten anderer Fahrzeuge berücksichtigt) sowie die Integration eines Navigationssystems vor. Damit adressiert Reichardt die aus der Literatur bekannten Defizite rein paralleler Basisarchitekturen (vgl. Kap. 3.2).

Insgesamt bleibt festzuhalten, dass trotz der geschilderten Bedenken hinsichtlich der Umsetzbarkeit der funktionalen Anforderungen die beiden Varianten (parallel-kooperativ, parallel-konkurrierend) nicht aus dem nachfolgenden Bewertungsprozess ausscheiden. Eine formale Falsifikation ist auch deswegen nicht möglich, da parallele Architekturen prinzipiell beliebig viele Verhalten besitzen können und die Arbitrierungsvorschrift beliebig komplex sein kann, sodass nicht ausgeschlossen werden kann, dass eine Anordnung von Verhalten und deren Koordination besteht, die genau die funktionalen Anforderungen dieser Arbeit erfüllen. Gestützt wird diese Vermutung von der Tatsache, dass das Team Carolo in der DARPA Urban Challenge zumindest für die Querführung des Fahrzeugs einen verhaltensbasierten Ansatz erfolgreich einsetzen konnte (vgl. Kap. 3.4.2). Vor der Bewertung der Varianten hinsichtlich der Erfüllung der nichtfunktionalen Anforderungen wird im nächsten Abschnitt zunächst die hybride Architekturvariante ermittelt.

4.1.3 Hybride Variante

Hybride Basisarchitekturen zeichnen sich durch die Kombination von reaktiven und deliberativen Systemkomponenten aus und verfügen gemäß Kap. 3.3 über mindestens zwei Hierarchieebenen bzw. Schichten, in die die beiden Komponenten abgebildet werden. Entsprechend dieses Grundsatzes werden die bereits in Abbildung 4-1 und Abbildung 4-2 gezeigten Systembausteine in eine beispielhafte hierarchische Grundordnung gebracht. Das Ergebnis dieses Schrittes ist in Abbildung 4-3 zu sehen.

²¹⁹ Reichardt (1996): Kontinuierliche Verhaltenssteuerung eines autonomen Fahrzeugs..., S. 147

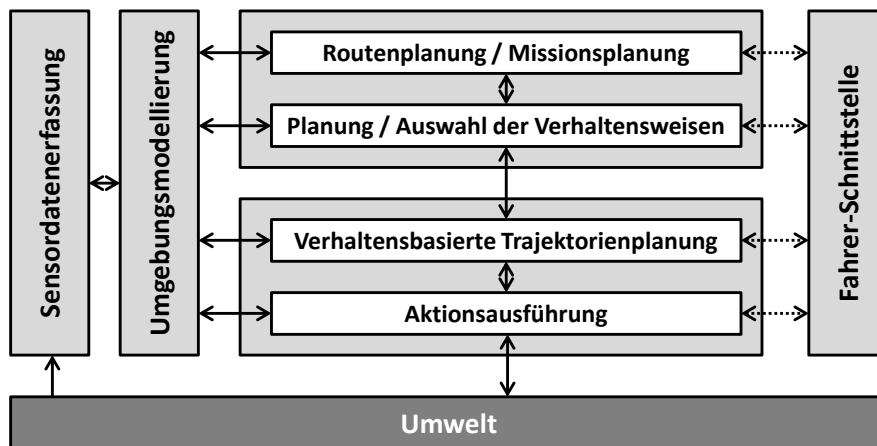


Abbildung 4-3: Grobstruktur der hybriden Architekturvariante

Die Trennung von deliberativen und reaktiven Komponenten ist dabei in Bildmitte durch die zwei graugefärbten Schichten angedeutet. Im oberen Teil sind die planenden Module repräsentiert, wobei in diesem Ausführungsbeispiel gemäß den funktionalen Anforderungen in die Routenplanung sowie die Planung bzw. Auswahl der Verhalten unterschieden wird. Hierbei ist es denkbar, dass der Routenplaner vor Erreichen eines Verkehrsknotens die Kante vorgibt, in die das Fahrzeug gemäß der Routenberechnung zur Erreichung des Ziels einfahren müsste. In einer niedrigeren Abstraktions- und Hierarchieebene erfolgt auf Basis dieser Information sowie unter Berücksichtigung der vorherrschenden Verkehrsregeln eine Auswahl bzw. auch eine Sequenzierung der dafür notwendigen Verhaltensweisen. In einer weiteren, untergeordneten Ebene werden die Verhaltensweisen, die im Rahmen der Anforderungsdefinition ermittelt wurden (vgl. Tabelle 2-1), implementiert. Bei Aktivierung der Verhaltensweisen würden dazu passende Fahrzeug-Trajektorien unter direkter Berücksichtigung der geometrischen, quasi-kontinuierlichen Umgebungsmodelle berechnet und somit die Reaktivität des Systems sichergestellt.

Jedes erwähnte deliberative bzw. reaktive Planungsmodul benötigt zur Aufgabenerfüllung spezifische Umgebungsinformationen mit einem darauf ausgerichteten Detaillierungs- bzw. Abstraktionsgrad. Diese Informationen werden durch das Modul der Umgebungsmodellierung zur Verfügung gestellt, wobei gemäß Matthaei²²⁰ in (absteigend von oben nach unten) topologische, semantische und geometrische Informationsmerkmale und ggf. benötigten Zwischenstufen unterschieden werden kann. Inspiriert durch die RCS-Architektur in Abbildung 3-6 wurde in Abbildung 4-3 ebenfalls eine horizontale Komponente „Fahrerschnittstelle“ eingeführt, wobei angedeutet wird, dass der Fahrer auf jeder Systemebene Eingriffsmöglichkeiten besitzt, wie sie z. B. in Form der Manöverdelegation auch in den funktionalen Anforderungen dieser Arbeit gefordert wird.

²²⁰ Matthaei (2015): Wahrnehmungsgestützte Lokalisierung in fahrstreifengenauen Karten..., S. 35 ff

Gemäß den Schlussfolgerungen aus Kap. 3.3 sowie den Anwendungsbeispielen aus Kap. 3.4 kann davon ausgegangen werden, dass eine hybride Architekturvariante prinzipiell in der Lage ist, die funktionalen Anforderungen dieser Arbeit zu erfüllen. Hinsichtlich der Laufzeiten des Kontrollzyklus ergeben sich analog zur parallelen Anordnung keine grundsätzlichen Bedenken, da die reaktiven Verhalten einen eigenen unterlagerten Kontrollzyklus bilden. Aus diesem Grund wird auch die hybride Basis-Architektur hinsichtlich der Erfüllung nichtfunktionaler Anforderungen überprüft.

4.2 Bewertung der Varianten und Auswahl der Basis-Architektur

Grundlage zur Bewertung der erarbeiteten Basis-Architekturvarianten stellt die in Kap. 2.3.3 vorgestellte Checkliste zur Überprüfung der nichtfunktionalen Anforderungen dar (siehe Tabelle 2-2). Das Ergebnis dieses Arbeitsschrittes ist in Tabelle 4-1 dargestellt, wobei im Folgenden die wichtigsten Aspekte zusammengefasst werden.

Das erste Kriterium, das der Forderung nach Testbarkeit zugeordnet werden kann, ist die Möglichkeit des Auftretens zyklischer Abhängigkeiten in den Modulstrukturen der drei Varianten *parallel-kooperativ* (V1), *parallel-konkurrierend* (V2) und *hybrid* (V3). Dabei wird untersucht, ob es zu logischen Kreisbezügen in der Architektur kommen kann, die *nicht* über die Regelstrecke Fahrzeug-Umwelt erklärt werden können. Prinzipbedingt besteht in der parallel-kooperativen Variante (V1) keine Gefahr zyklischer Abhängigkeiten, da die implementierten Verhaltensweisen gänzlich unabhängig voneinander implementiert sind. Dies steht im Gegensatz zu Variante 2, da je nach Ausprägung des Inhibitionsnetzwerkes nicht ausgeschlossen werden kann, dass sich Verhalten gegenseitig aktiv beeinflussen. Ein gegenseitiges, bewusstes Hemmen zweier Verhaltensweisen wäre eine mögliche Konsequenz von logischen Kreisbezügen zwischen Modulen in parallel-konkurrierenden Architekturen.

Auch bei hybriden Architekturen kann es zu zyklischen Abhängigkeiten kommen, wenn beispielsweise die Aktivierung / Deaktivierung eines Verhaltens in Planungsmodulen rein auf der Rückmeldung des entsprechenden Verhaltens basiert²²¹.

²²¹ Bsp.: Ein Planungsmodul beauftragt einen Fahrstreifenwechsel, der erst dann als beendet erklärt wird, wenn das Verhalten die Ausführung bestätigt. Wird aufgrund einer externen Störung (z. B. eine bleibende Querablage im Fahrstreifen durch Spurrillen) oder aufgrund eines Implementierungsfehlers diese Bedingung nicht erfüllt, so hängt das Verhalten in einer „Endlosschleife“ fest.

Tabelle 4-1: Bewertung der Basis-Architektur-Varianten

Nr.	Kriterium	V1: Parallel-Kooperativ	V2: Parallel-Konkurrierend	V3: Hybrid
1	Prüfung auf zyklische Abhängigkeiten (Kreisbezüge)	Nicht vorhanden, da Verhaltensweisen unabhängig voneinander sind.	Möglich, wenn die Aktivität eines Verhaltens von der geforderten Aktivität eines anderen Verhaltens abhängig ist und umgekehrt (z. B. gegenseitiges Hemmen in einem Inhibitionsnetzwerk).	Möglich, wenn die Aktivierung eines Verhaltens rein auf der Rückmeldung des zu aktivierenden Verhaltens abhängt (z. B. wenn das Verhalten in einer Endlosschleife läuft).
2	Prüfung auf Determinismus	Nicht-deterministisch.	Weitestgehend nicht-deterministisch, je nach Auslegung der Inhibitionsvorschrift.	Weitestgehend deterministisch (wenn höhere Hierarchieebenen regelbasiert implementiert werden).
3	Prüfung auf Möglichkeit der Dekomposition	Keine Trennung von Entscheidung und Ausführung, da Entscheidungen nicht explizit repräsentiert sind.		Trennung von Entscheidung und Ausführung durch Hierarchieebenen möglich.
4	Prüfung auf hohe Kohäsion	Hohe Kohäsion auf Modulebene, verminderte Kohäsion auf Systemebene		Hohe Kohäsion auf Modul- und Systemebene.
5	Prüfung auf lose Kopplung	Lose Kopplung auf Modulebene, jedoch wird ein erhöhtes Verständnis der Koordinationsvorschrift (z. B. Gewichtungungen) vorausgesetzt, um Ziele auf Systemebene zu erreichen.	Starke Kopplung, tiefgehende Kenntnis der Koordinationsvorschrift notwendig (z. B. Inhibitionsnetzwerk).	Weitestgehend lose Kopplung, da Entscheidungen und Ausführungen in definiertem Spielraum unabhängig voneinander geändert werden können.
6	Prüfung auf Einhaltung des Information-Hiding-Prinzips	Keine Schichten bzw. Abstraktionsebenen.		Schichten bzw. Abstraktionsebenen vorhanden und an Anforderungen anpassbar.
7	Prüfung auf Einhaltung des Information-Hiding-Prinzips	Keine Bewertung möglich, da keine Abstraktionsebenen vorhanden.		Noch keine Bewertung möglich, da Schnittstellen noch nicht definiert sind.
8	Prüfung auf Separation-of-Concerns	Hohe Separation-of-Concerns, da die Verhaltensweisen und auch die Zielführung als eigenständige Module implementiert werden können.		
9	Prüfung auf Separation-of-Concerns	Hohe Separation-of-Concerns, da die Hardwarearchitektur, z. B. eingesetzte Umfeldsensoren oder MMS-Elemente keinen Einfluss auf die Basis-Architektur besitzen.		
10	Prüfung auf externe Unabhängigkeit	Änderungen in der Hardware-Architektur wirken sich nicht direkt auf die implementierten Verhaltensweisen aus, sondern nur auf die Sensordatenerfassung und Aktionsausführung.		

Tabelle 4-1: Bewertung der Basis-Architektur-Varianten (Fortsetzung)

Nr.	Kriterium	V1: Parallel-Kooperativ	V2: Parallel-Konkurrierend	V3: Hybrid
11	Prüfung der Modulgröße	Kriterium zur Bewertung der Basis-Architektur nicht geeignet.		
12	Prüfung auf mögliche Refaktorisierung	Keine Redundanzen auf Konkretisierungsebene der Basis-Architektur feststellbar. Widersprüche zwischen Verhalten möglich.	Keine Redundanzen auf Konkretisierungsebene der Basis-Architektur feststellbar. Widersprüche zwischen den Verhalten können auf deliberativer Ebene adressiert werden.	
13	Prüfung auf semantische Kohäsion	Modulfunktionalitäten lassen sich semantisch weitestgehend einfach und eindeutig beschreiben.	Modulfunktionalitäten lassen sich semantisch weitestgehend einfach und eindeutig beschreiben, jedoch aufgrund des Zusammenspiels der Ebenen komplexer.	

Am Beispiel der Prüfung auf zyklische Abhängigkeiten wird deutlich, dass insbesondere bei parallel-konkurrierenden und hybriden Architekturvarianten ein höherer Testaufwand betrieben werden muss, um logische Kreisbezüge und „Endlosschleifen“ im Informationsaustausch der Module ausschließen zu können. Dies stellt jedoch nur scheinbar ein Vorteil für V1 dar, denn schließlich kann auch hier ein gegenseitiges Hemmen von Verhalten auftreten²²², das jedoch in Gegensatz zu V2 und V3 womöglich nur durch eine aufwändige Abstimmung und Parametrierung der Verhaltensweisen zu lösen wäre, was ebenfalls einen hohen Implementierungs- und Testaufwand nach sich ziehen würde.

Dieser Aspekt spiegelt sich ebenfalls im zweiten Kriterium wider, der Forderung nach Erreichung eines hohen Grades an Determinismus und somit nach Vorhersagbarkeit des Systemverhaltens. Ein weitestgehend unvorhersagbares Systemverhalten der parallelen Architekturvarianten 1 und 2 führt nicht nur aus funktionaler Sicht im Kontext einer Verkehrsregel-bestimmten Einsatzdomäne zu Problemen, wie u.a. am Beispiel der DAMN-Architektur im Stadtpilot-Projekt belegt werden kann (vgl. Kap. 3.4.3), sondern führt auch zu einem erhöhten Testaufwand, um beispielsweise verkehrsregelkonformes Verhalten in verschiedensten Szenerie- und Verkehrsobjektkonstellationen nachweisen zu können. Durch dezidierte Planungselemente in der hybriden Architekturvariante kann

²²² Ein klassisches Beispiel hierfür ist das Auffahren auf ein Hindernis, wobei sich die Verhalten „links Ausweichen“ und „rechts Ausweichen“ gegenseitig aufheben.

hier ein höherer Grad an Determinismus und somit ein geringerer Testaufwand erreicht werden. Dabei wird das Testen auch dadurch erleichtert, dass bei hybriden Systemen in die Entscheidung über und Ausführung von Verhaltensweisen unterschieden werden kann, was es ermöglicht, diese beiden Elemente unabhängig voneinander zu implementieren, zu testen (Kriterium 3: Möglichkeit der Dekomposition) und zu ändern bzw. parametrisieren. Letzterer Aspekt betrifft ebenfalls die Forderung nach einer losen Modulkopplung (Kriterium 5), die in Variante 1 zwar auf Modulebene gegeben ist, jedoch eine Änderung der Verhaltensweisen implizite und nur schwer vorhersagbare Auswirkung auf das resultierende Gesamtsystemverhalten besitzt. Änderungen von Verhaltensweisen in Variante 2 setzen darüber hinaus eine genaue Kenntnis der Inhibitions- bzw. Arbitrierungsvorschrift voraus, was zu einer starken Modulkopplung führt und die Änderbarkeit bzw. Erweiterbarkeit einer solchen Architektur erschwert.

Alle drei Varianten zeichnen sich durch eine hohe Separation-of-Concerns gemäß den Kriterien 8 und 9 und sowie durch eine geringe Abhängigkeit von externen Hardwareelementen (Sensoren, MMS-Elemente) gemäß Kriterium 10 aus. Hinsichtlich möglicher Redundanzen der Zuständigkeiten (Kriterium 12) konnten auf der Konkretisierungsebene der Basis-Architektur keine Bedenken gefunden werden, jedoch besteht bei den Varianten 1 und 2 die Gefahr widersprüchlicher Verhaltensweisen, die in der hybriden Architekturvariante im deliberativen Anteil gezielt adressiert werden können. Die Zuständigkeiten der entsprechenden Module der drei Varianten lassen sich semantisch weitestgehend einfach beschreiben (Kriterium 13), wobei diese Aufgabe in Variante 3 durch die Einbringung der (Hierarchie-) Ebenen komplexer zu sein scheint.

Nach Prüfung der drei erarbeiteten Architekturvarianten hinsichtlich der Erfüllung nichtfunktionaler Anforderungen mit den Schwerpunkten der Testbarkeit sowie Änderbarkeit bzw. Erweiterbarkeit wird die Variante 3 als Basisarchitektur für die in dieser Arbeit zugrundeliegenden Aufgabenstellung präferiert. Insbesondere die mangelnde Planbarkeit der Verhaltensweisen sowie die in der Literatur häufig beschriebenen komplexen Arbitrierungsmechanismen paralleler Basisarchitekturen können zu einer verringerten Vorhersagbarkeit des Systemverhaltens führen, das sich - zumindest im Umfang der funktionalen Anforderungen dieser Arbeit - in einem erhöhten Test- und Abstimmungsaufwand bei Änderungen / Erweiterungen widerspiegelt. Aus diesen Gründen wird im weiteren Verlauf der Arbeit die hybride Architekturvariante weiter detailliert. Die Herausforderung dabei wird sein, die deliberativen und reaktiven Systemelemente so miteinander zu verknüpfen, dass ein möglichst widerspruchs- und redundanzfreies Zusammenspiel zwischen Verhaltensentscheidung und -ausführung entsteht, das sich semantisch einfach und eindeutig beschreiben lässt.

Obgleich insgesamt durch die Unterteilung in serielle, parallele und hybride Basisarchitekturen der Lösungsraum gemäß Literatur vollständig beschrieben ist, kann nicht ausgeschlossen werden, dass andere Strukturen existieren, die die funktionalen Anforde-

rungen der Arbeit umsetzen können, z. B. monolithische Strukturen oder künstliche (neuronale) Netze. Es kann jedoch vermutet werden, dass solche Varianten hinsichtlich der Erfüllung der nichtfunktionalen Anforderungen Nachteile besitzen, insbesondere im Kontext der einfachen semantischen Beschreibbarkeit sowie der Modularität als Obermenge der o.g. Anforderungen, die eine große Rolle in der Teamarbeit und Arbeitsteilung der beteiligten Interessensgruppen spielen und bereits in Normen wie der ISO 26262 manifestiert sind (vgl. Kap. 2.3.2).

5 Entwicklung der Referenzarchitektur

Im vorliegenden Kapitel wird auf Basis der ausgewählten hybriden Basisstruktur eine Referenzarchitektur entwickelt, die eine architektonische Gesamtlösung für den Problembereich des automatisierten Fahrens mit Fahrereinbindung darstellen soll. Dabei wird im Folgenden die Architektur gemäß dem in Kap. 1.5 beschriebenen Vorgehen mit Hilfe der Problemfaktorisierung schrittweise konkretisiert.

Im ersten Schritt (Abschn. 5.1) erfolgt eine Festlegung der Systemhierarchie, die die Zuständigkeiten der Ebenen maßgebend prägt. In Abschn. 5.2 werden darauf aufbauend die Aufgaben der planenden und ausführenden Module definiert, um die in Kap. 2.2.3 beschriebenen funktionalen Anforderungen umsetzen zu können. Ist dies erfolgt, so lässt sich der sich daraus ergebende Informationsbedarf der einzelnen Module ermitteln, der seitens der Sensordatenverarbeitung und Umfeldmodellierung gedeckt werden muss (s. Abschn. 5.4).

5.1 Festlegung der Systemhierarchie

5.1.1 Hierarchische Abstraktionsmodelle im Stand der Technik

Als Strukturierungshilfe zur Erstellung hierarchischer Mehrebenensysteme nutzen Systemarchitekten für automatisierte Fahrzeugführungssysteme häufig theoretische Abstraktionsmodelle. Ein bekanntes und häufig genutztes Beispiel hierfür stellt das Modell für zielgerichtetes menschliches Handeln nach Rasmussen²²³ dar. Bei der Nutzung dieses Modells zur Strukturierung von Architekturen wird entsprechend der Mensch als Vorbild für ein autonomes System herangezogen²²⁴. Beispiele für die Anwendung des Modells sind u.a. in Dickmanns²²⁵ (4-D-Ansatz) und Maurer²²⁶ gegeben (siehe Kap. 3.4.1). Rasmussen beschreibt drei hierarchisch angeordnete Abstraktionsebenen: Die *fertigkeitsbasierte*, *regelbasierte* und *wissensbasierte* Ebene (siehe Abbildung 5-1, links). Jede Ebene beschreibt dabei eine unterschiedliche kognitive Inan-

²²³ Rasmussen (1983): Skills, Rules and Knowledge...

²²⁴ siehe z. B. Maurer (2000): Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen, S. 27

²²⁵ Dickmanns et al. (1994): The Seeing Passenger Car „VaMoRs-P”

²²⁶ Maurer (2000): Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen

spruchnahme des Menschen bei der Bewältigung von alltäglichen Routineaufgaben bis hin zu völlig neuen und unbekannten Situationen.

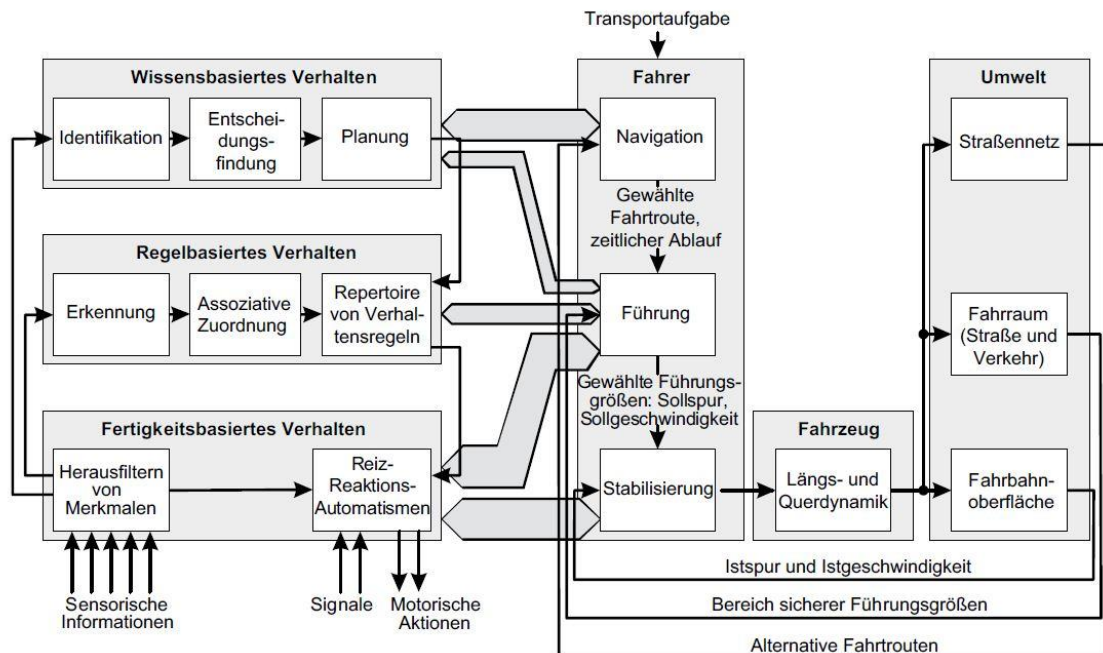


Abbildung 5-1: Zusammenhang zwischen den Abstraktionsmodellen von Rasmussen (links) und Donges (rechts)²²⁷

Auf der untersten Ebene beschreibt das fertigkeitsbasierte Verhalten eine sehr schnelle und reflexartige Wandlung von sensorischen Informationen in motorische Aktionen. Ein solches Verhalten tritt auf, wenn Abläufe menschlichen Handelns durch häufige Exposition so trainiert und erlernt wurden, dass diese in einem stetigen Fluss ablaufen und keine bewusste Kontrolle des Menschen erfordern²²⁸. Regelbasiertes Verhalten auf der zweiten Ebene tritt dann auf, wenn der Mensch mit Aufgaben konfrontiert wird, die ihm bereits durch vorherige Erfahrungen bekannt sind. Nach Erkennen der Situation kann der Mensch deswegen auf ein selbst erlerntes Repertoire von Verhaltensregeln zurückgreifen, wobei die aus subjektiver Sicht effektivste Regel als Vorsteuerung für das fertigkeitsbasierte Verhalten ausgewählt wird. Auf der dritten Ebene ist das wissensbasierte Verhalten repräsentiert. Dabei wird der Mensch einer ihm unbekannten bzw. unvorhergesehenen Situation ausgesetzt, zu dem er (noch) keine Verhaltensregel zuordnen kann, sodass er mental verschiedene Handlungsalternativen erarbeiten und gegeneinander abwägen muss. Entsprechend nimmt der Vorgang wissensbasierten Verhaltens mehr Zeit und kognitive Ressourcen in Anspruch als regelbasiertes Verhalten (analog verhält es sich mit dem Zusammenhang zwischen regel- und fertigkeitsbasiertem Verhalten).

²²⁷ Entnommen aus Donges (2015): Fahrerverhaltensmodelle, S. 19

²²⁸ vgl. Donges (2015): Fahrerverhaltensmodelle, S. 18

Wird das Modell nach Rasmussen auf die Aufgabe des Führens eines Fahrzeugs übertragen, so wären Beispiele für fertigkeitsbasiertes Verhalten das manuelle Kuppeln und Schalten eines Ganges, für regelbasiertes Verhalten die korrekte Reaktion auf Vorfahrtsregelungen wie z. B. das Anhalten vor einem roten Lichtsignal und für wissensbasiertes Verhalten das Navigieren in einem unbekannten Umfeld. Dabei ist zu beachten, dass eine solche Zuordnung erfahrungsabhängig ist. So wäre es denkbar, dass die aufgezählten Beispiele für einen Fahranfänger womöglich zunächst alle wissensbasiert ablaufen.

Ein zweites, häufig genutztes²²⁹ Abstraktionsmodell stellt die 3-Ebenen-Hierarchie der Fahrzeugführung nach Donges^{230a} dar (siehe Abbildung 5-1, rechts), die die Fahraufgabe in die drei Ebenen der *Navigation*, der (Bahn-) *Führung* sowie der *Stabilisierung* untergliedert. Bei vorgegebener Transportaufgabe findet gemäß dem Modell zunächst die Lösung der Navigationsaufgabe statt, wobei eine Fahrtroute aus dem vorhandenen Straßennetz unter Berücksichtigung der Fahrzeit ausgewählt wird. Donges bezeichnet diese Teilaufgabe als einen zeitlich diskreten Prozess, bei dem die Einhaltung der Route örtlich punktuell anhand markanter Streckenmerkmale überwacht wird^{230b}. Auf der zweiten Ebene findet die Führungsaufgabe statt, bei der unter Berücksichtigung der sich ständig ändernden Verkehrssituation und des geplanten Fahrtablaufs Führungsgrößen, wie z. B. die Sollgeschwindigkeit oder der Sollspur²³¹, als antizipatorische Vorsteuerung (open loop) ausgewählt werden. Auf der Ebene der Stabilisierung findet schließlich die kompensatorische Regelung (closed loop) auf Fahrdynamikebene statt.

Beispiele für die Nutzung des Modells nach Donges zur Strukturierung hierarchischer Systemarchitekturen für automatisierte Fahrzeuge sind u.a. in Geyer²³² für das manöverbasierte Automationskonzept CbW (vgl. Kap. 3.4.4), Löper²³³ und im weiteren Sinne auch in der Architektur des DARPA-Fahrzeugs „BOSS“ (vgl. Kap. 3.4.2) und Matthaei²³⁴ gegeben.

Der Zusammenhang der Modelle von Rasmussen und Donges wird in Abbildung 5-1 visualisiert: So werden die drei Ebenen der Modelle zunächst direkt gegenübergestellt, sodass der Eindruck entsteht, dass die Navigationsaufgabe vorwiegend wissensbasiertes, die Führungsaufgabe regelbasiertes und die Stabilisierungsaufgabe fertigkeitsbasiertes Verhalten erfordert. Tatsächlich lassen sich jedoch Beispiele finden, die eine solche

²²⁹ vgl. Matthaei et al. (2015): Autonomes Fahren, S. 1153

²³⁰ Donges (2015): Fahrerverhaltensmodelle. a: -; b: S. 19

²³¹ Unter der Sollspur wird in diesem Kontext die Bahn des Fahrzeugs verstanden, die sich an den Struktur-Elementen der Fahrstreifen orientieren, vgl. Donges (2015): Fahrerverhaltensmodelle, S. 22

²³² Geyer (2013): Maneuver-Based Vehicle Guidance Based on the Conduct-by-Wire Principle, S. 114

²³³ Löper (2008): Kooperative, manöverbasierte Automation und Arbitrierung...

²³⁴ Matthaei (2015): Wahrnehmungsgestützte Lokalisierung in fahrstreifengenauen Karten..., S. 23

Aufteilung falsifizieren, wie u.a. in Matthaei²³⁵ dargestellt ist und in vorherigem Beispiel des Fahranfängers ausgeführt wurde. Aus diesem Grund existieren in Abbildung 5-1 Kreuzverbindungen zwischen den Ebenen der unterschiedlichen Modelle, was auch bedeutet, dass es sich bei den beiden Modellen nicht um konkurrierende bzw. sich widersprechende Ansätze handelt, sondern zwei unterschiedliche Perspektiven im Kontext der Strukturierung automatisierter Fahrzeugführungssysteme darstellen. Dennoch stellt sich für den Systemarchitekt die Frage, welches Abstraktionsmodell eine größere Hilfestellung zur Definition der Anzahl der Hierarchieebenen und deren funktionalen Zuständigkeiten bei hybriden Systemen bietet, was im Folgenden näher untersucht wird.

Die Existenz weiterer Abstraktionsmodelle, anhand derer sich eine mögliche Systemhierarchie ableiten lässt, kann nicht ausgeschlossen werden. Im Rahmen der durchgeführten Literaturrecherche im Bereich der Robotik, des Standes der Technik von A/A-Systemen sowie der Automobilergonomie²³⁶ konnten jedoch keine weitere sich signifikant unterscheidbare²³⁷ Modelle identifiziert werden, sodass die folgenden Ausführungen auf den Modellen von Rasmussen und Donges aufbauen.

5.1.2 Einflussfaktoren auf die Systemebenen

Laut Antsaklis & Passino²³⁸ ist die Anzahl der Ebenen einer funktionalen Architektur anwendungsabhängig, wird jedoch maßgeblich von vier Faktoren beeinflusst:

1. Vom Umfang bzw. von der Art, in der ein (menschlicher) Operator in das System eingreifen kann
2. Vom Grad der Autonomie des Systems
3. Vom funktionalen Umfang der Subsysteme
4. Von der hierarchischen Charakteristik der Umwelt des Systems

Bereits der erste Faktor liefert wertvolle Hinweise zur Strukturierung des Gesamtsystems durch Gegenüberstellung der in Kap. 2.2 definierten funktionalen Anforderungen. Die in dieser Arbeit zu entwickelnde Architektur soll verschiedene Automationsgrade ermöglichen, wobei jeweils unterschiedliche Eingriffsmöglichkeiten des menschlichen Operators, in diesem Fall des Fahrers, vorgegeben sind. Anhand der entsprechenden

²³⁵ Matthaei (2015): Wahrnehmungsgestützte Lokalisierung in fahrstreifengenaue Karten..., S. 23

²³⁶ vgl. Bubb et al. (2015): Automobilergonomie

²³⁷ Bubb beschreibt bspw. eine „Hierarchie der Fahraufgabe“ auf Basis einer Einteilung von Geiser (1985), die sich in die primäre, sekundäre und tertiäre Aufgabe gliedert. Jedoch entspricht die primäre Aufgabe dem Modell nach Donges (Navigation, Führung, Stabilisierung), die sekundäre Aufgabe stellen Aktionen wie z. B. das Betätigen der Scheibenwischer dar und die tertiäre Aufgabe sind fahrfremde Tätigkeiten wie z. B. das Bedienen des Radios; Bubb (2015): Automobilergonomie, S. 20 ff

²³⁸ Antsaklis & Passino (1992): Introduction to Intelligent Control Systems..., S. 7

Kategorisierung in Tabelle 5-1 wird deutlich, dass den drei Automationsgraden auch drei Ebenen der Fahrereingriffsmöglichkeiten zugeordnet werden können. Im vollautomatisierten Modus gibt der Fahrer analog zu im heutigen Stand der Technik erhältlichen Navigationssystemen ein Routenziel sowie ggf. Routenoptionen (schnellste / kürzeste Strecke etc.) vor²³⁹.

Tabelle 5-1: Automationsgrade und Einflussmöglichkeiten des menschlichen Operators

Automationsgrad ²⁴⁰	Einflussmöglichkeiten des Fahrers
Vollautomatisiert	Eingabe des Navigationsziels
Teilautomatisiert	Delegation von Fahrmanövern
Assistiert	Kontrolle der Fahrzeuglängs- und Querverführung

Im teilautomatisierten Modus delegiert der Fahrer stattdessen Fahrmanöver an die Automation, die diese dann selbstständig ausführt. Schreiber²⁴¹ führt zur Definition von Fahrmanövern im Kontext des teilautomatisierten Fahrens auf Autobahnen eine umfassende Studie durch und vergleicht verschiedene Quellen, die Diskretisierungsansätze zur Einteilung von Fahrmissionen darstellen, wobei an dieser Stelle insbesondere auf Tölle²⁴² und Fastenmeier²⁴³ hingewiesen sei. Als Anforderung an eine MMS auf Manöverebene werden von Schreiber für den Fahrer verständliche Fahrmissionsabschnitte (Handlungsabschnitte) gefordert, die sequentiell aneinandergereiht die Fahrmission abbilden können. Als Ergebnis der Studie führt Schreiber die Manöver „Anhalten“, „Anfahren“ sowie Fahrstreifenwechsel (links / rechts) auf. Eine weitere Erkenntnis der Studie ist, dass Fahrer Manöver wie z. B. „der Straße folgen“, „einem Fahrzeug folgen“ oder „einem Fahrzeug annähern“ nicht als Manöver interpretieren. Deshalb unterscheidet Schreiber bei der Begriffsdefinition zwischen expliziten Manövern (ein in sich abgeschlossener Teil der Bahnführungsaufgabe mit definierten Anfangs- und Endbedingungen, der vom Fahrer beauftragt wird) und impliziten Manövern (nicht zwingend abgeschlossener Teil einer Bahnführungsaufgabe, der durch explizite Fahrmanöver oder durch das Fahrmissionsende beendet und nicht vom Fahrer beauftragt wird). Bezugnehmend auf die vorliegende Arbeit kann demnach festgestellt werden, dass die in Tabelle 2-1 identifizierten Verhaltensweisen auch eine geeignete Schnittstelle zur Ma-

²³⁹ siehe dazu z. B. Kleine-Besten et al. (2015): Navigation und Verkehrstelematik, S. 1053

²⁴⁰ vgl. Kap. 2.1

²⁴¹ Schreiber (2011):Konzeptionierung und Evaluierung eines Ansatzes..., S. 41 ff

²⁴² Tölle (1996):Ein Fahrmanöverkonzept für einen maschinellen Kopiloten

²⁴³ Fastenmeier (1995):Autofahrer und Verkehrssituation

növerdelegation darstellen, da sie mit der Definition von Schreiber vereinbar sind²⁴⁴. Dabei ist die in Tabelle 2-1 definierte Verhaltensweise 1 (Fahrstreifen befahren) als implizites Fahrmanöver und alle anderen Verhaltensweisen als explizite Manöver anzusehen.

Im assistierten Modus (im Kontext von PRORETA 3, vgl. Kap. 2.1) ist der Fahrer selbst für die Fahrzeuglängs- und Querverführung zuständig, wobei das System analog zu unfallvermeidenden bzw. unfallfolgenmindernden Fahrerassistenzsystemen die aktuelle Verkehrssituation überwacht und im Falle einer kritischen Situation warnt und eingreift²⁴⁵.

Der zweite o.g. Einflussfaktor stellt der Grad der Autonomie des zu entwickelnden Systems dar. Ein anschauliches Beispiel, dass der Grad der Autonomie Einfluss auf die Anzahl der Ebenen besitzen kann, wurde bereits im Beispiel der CbW-Systemarchitektur aufgezeigt (vgl. Kap. 3.4.4), in der aufgrund der Teilautomatisierung die (oberste) Navigationsebene entfällt (bzw. dem Fahrer vorbehalten ist) und somit nur über zwei Ebenen verfügt. Eine weitreichendere Betrachtung dieses Aspekts stellen Bayouth et al.²⁴⁶ an, die eine Architektur mit drei Ebenen analog zur Struktur von Donges vorstellen. Dabei werden in einem Gedankenexperiment die Zuständigkeiten der Ebenen zwischen Fahrer und Automation aufgeteilt und variiert. So sind auch Systeme mit nur einer Ebene denkbar: Wird nur die Navigationsebene automatisiert, so handelt es sich bspw. um ein konventionelles Navigationsgerät. Anhand dieser Beispiele ist offensichtlich, dass ein höherer Grad an Autonomie - in diesem Fall bezogen auf den Automationsgrad - mit einer erhöhten Anzahl von Systemebenen verbunden ist, sofern das gleiche Abstraktionsmodell zugrunde gelegt wird (vgl. auch RCS-Architektur, s. Kap. 3.3).

Der dritte o.g. Faktor, der die Anzahl der Systemebenen beeinflusst, ist der funktionale Umfang der Subsysteme. Dies bedeutet, dass die Anzahl der Ebenen mit deren Funktionsumfang in einem wechselseitigen Verhältnis stehen. Siedersberger²⁴⁷ untergliedert beispielsweise Fahrmanöver in seinem Fähigkeitenkonzept weiter in eine Handlungsebene (das eigentliche Manöver) und eine Fertigkeitenebene, die Handlungsprimitive wie z. B. das Verzögern, Beschleunigen und Anhalten umfassen. Ein Fähigkeitsnetz bestimmt dabei, auf welche Handlungsfertigkeiten die Manöver zugreifen können, um

²⁴⁴ Dies zeigt sich auch darin, dass die von Schreiber identifizierten Manöver für Autobahnfahrten den Verhaltensweisen in Anhang A.1 (Autobahn) entsprechen, wobei im Falle dieser Arbeit das Manöver „Anfahren“ nicht berücksichtigt wird, da von einer „fliegenden“ Aktivierung des Systems, d.h. während es sich bereits im Straßenverkehr bewegt, ausgegangen wird (Ein- und Ausparkmanöver sind nicht Teil der funktionalen Anforderungen).

²⁴⁵ s. z. B. Winner (2015): Grundlagen von Frontkollisionsschutzsystemen

²⁴⁶ Bayouth et al. (1997): Hybrid Human-Computer Autonomous Vehicle Architecture

²⁴⁷ Siedersberger (2003): Komponenten zur automatischen Fahrzeugführung..., S. 95 ff

ihr Ziel zu erreichen. Wird diese Hierarchie innerhalb der Verhaltensweisen ebenfalls in einer zusätzlichen Systemebene der Architektur abgebildet, so entstehen aus einer Ebene mit einem integralen Funktionsumfang zwei Ebenen mit verringertem Funktionsumfang. Im Rahmen des vorliegenden Kapitels wird jedoch zunächst versucht, die minimale notwendige Anzahl von Systemebenen zu identifizieren, da eine weitere Detaillierung der Ebenen, sofern notwendig, auch im späteren Verlauf der Arbeit möglich ist.

Der vierte Faktor, der maßgebend hinsichtlich der Systemstruktur ist, stellt die hierarchische Charakteristik der Umwelt des Systems dar. Dass sich die Umwelt des Straßensystems hierarchisch beschreiben lässt, zeigt die Art und Weise, wie digitale Straßenkarten aufgebaut sind. So sind beispielsweise in OpenStreetMap Straßentypen (z. B. Autobahn, Landstraße etc.) als Attribute von allgemeinen Straßensegmenten repräsentiert²⁴⁸. Straßentypen besitzen wiederum ein Attribut „Fahrstreifen“, die wiederum über Attribute zur Fahrtrichtung oder zu geometrischen Informationen, wie z. B. der Breite des Fahrstreifens, verfügen können. Attributs-intern lassen sich teilweise orthogonal zur Datenstruktur ebenfalls semantische Hierarchien bilden, z. B. dass einer Autobahn eine höhere Hierarchiestufe zugerechnet wird als einer Stadtstraße, was bei der effizienten Routenberechnung eine wichtige Rolle spielt²⁴⁹. Eine ausführliche, hierarchisch aufgebaute Klassenstruktur zur Beschreibung einer digitalen Karte auf Basis von OpenStreetMap wird in Schönichen²⁵⁰ beschrieben. Insbesondere auf Basis der in Kap. 2.2.2 vorgestellten Straßenbaurichtlinien lassen sich drei Gliederungsebenen der Systemumwelt eines automatisierten Fahrzeugs unterscheiden (vgl. auch Abbildung 5-1, rechts):

- Das Straßennetz in Form eines Graphen
- Die Anordnung der Strukturelemente, insbesondere der Fahrstreifen
- Detaillierte geometrischen Informationen zur Beschaffenheit der Strukturelemente wie z. B. Fahrbahnmarkierungen, Krümmungen / Neigungen

Im nächsten Schritt werden die in diesem Abschnitt gewonnenen Erkenntnisse den im vorherigen Abschnitt vorgestellten Abstraktionsmodellen gegenübergestellt und die Systemhierarchie festgelegt.

5.1.3 Identifikation der Systemebenen

Wird das Abstraktionsmodell nach Donges als Ausgangspunkt für die Strukturierung der hybriden Architektur zugrunde gelegt, so ergibt sich eine hohe semantische Übertrag-

²⁴⁸ s. http://wiki.openstreetmap.org/wiki/DE:Map_Features, letzter Zugriff am 03.10.16

²⁴⁹ Kleine-Besten et al. (2015): Navigation und Verkehrstelematik, S. 1054

²⁵⁰ Schönichen (2012): Integration von Informationen digitaler Karten in ein Fusions-Framework..., S. 28

barkeit mit den im vorherigen Kapitel genannten Einflussfaktoren. Dies wird in Tabelle 5-2 (linker Teil) deutlich. So können die Einflussmöglichkeiten des Operators in einen eindeutigen Zusammenhang mit den drei Ebenen nach Donges gebracht werden. Hinsichtlich der Autonomie des Systems lassen sich die Automationsgrade einer schrittweise hinzukommenden Automatisierung der nächsthöheren Abstraktionsebene nach Donges erklären, wie bereits im vorherigen Abschnitt beschrieben wurde²⁵¹. Dies gilt ebenso für die hierarchische Charakteristik der Systemumwelt, in der bspw. die Navigationsaufgabe nach Donges eine Kenntnis über das Straßennetz erfordert. Für die Bahnführungsaufgabe sind nicht ausschließlich, aber insbesondere die Kenntnis über das Vorhandensein und Art der Strukturelemente „Fahrstreifen“ von besonderem Interesse sowie für die Stabilisierungsaufgabe die genaue geometrische Beschreibung dieser²⁵².

Wird das Abstraktionsmodell menschlichen Handelns nach Rasmussen zugrunde gelegt (siehe Tabelle 5-2, rechter Teil), so lassen sich nicht alle Einflussfaktoren eindeutig adressieren. Ob bspw. die Eingabe eines Navigationsziels oder die Vorgabe von Fahrmanövern dem fertigungs-, regel-, oder wissensbasierten Verhalten zuzuordnen sind, hängt maßgeblich von der Erfahrung des Operators mit dem Bediensystem ab. Umgekehrt lassen sich die drei Ebenen nicht direkt einer Systemfunktion zuordnen, die eine menschliche Eingabe voraussetzen. Für den Einflussfaktor der Automationsgrade kann insoweit festgestellt werden, dass ein automatisiertes Fahrzeugführungssystem über fertigungs-, regel-, oder wissensbasiertes Verhalten, d.h. über schnelle und reflexartige Reaktionen, die kein komplexes semantisches Verständnis der Umwelt voraussetzt, verfügen muss (z. B. das Abstandhalten zu einem vorausfahrenden Fahrzeug). Dies entspricht der Definition von reaktiven Verhalten, die auch hybride Architekturen beinhalten (vgl. Kap. 4.1.3). Ebenfalls muss ein teil- bzw. vollautomatisiertes Fahrzeugführungssystem über regelbasiertes Verhalten verfügen, bei dem Symbole aus der Umwelt einer Verhaltensregel zugeordnet werden. Dies trifft insbesondere für Verkehrsregeln und die dafür in der StVO²⁵³ definierten Symbole in Form von (z. B. vorfahrtsregelnden) Verkehrszeichen oder Lichtsignalanlagen zu. Dass ein automatisiertes System, unabhängig vom Grad der Autonomie, über wissensbasiertes Verhalten verfügen muss, scheint jedoch nicht der Fall zu sein. So fasst Maurer²⁵⁴ zusammen, dass „derzeit [Anm.: Jahr 2000] keine algorithmischen

²⁵¹ Je nach Assistenzkonzept wäre die Bahnführung ebenfalls als Teil des assistierten Fahrens anzusehen (wenn eine Bahn- / Trajektorienplanung ausgeführt wird)

²⁵² Hierbei sei angemerkt, dass je nach Kopplung der Ebenen (s. nächster Abschnitt) weitere Informationen, z. B. über andere Verkehrsobjekte, für die Bahnführungs- und Stabilisierungsaufgabe relevant ist. Die Informationen über die Strukturelemente ist somit als notwendig aber nicht hinreichend zu klassifizieren.

²⁵³ Bundesministerium für Verkehr (2013): Straßenverkehrsordnung, Anlagen 1-4

²⁵⁴ Maurer (2000): Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen, S. 46

Analogien in der Technik zu den menschlichen wissensbasierten Methoden zur Ableitung der Handlungsregeln bekannt“ und laut Hertzberg & Chatila auch nicht notwendig sind²⁵⁵. Dies ist laut Maurer auch darin begründet, dass das Erlernen (und das „nicht wieder Vergessen“) neuer Regeln auf Basis unbekannter Situationen für den Menschen ein kognitiv mühsamer Prozess ist, aber technische Systeme im Gegensatz dazu nahezu beliebig viele Regeln und Verhaltensmuster verwalten können.

Wissensbasiertes Verhalten wird auch im Zusammenhang mit den Begriffen *künstliche Intelligenz* und *maschinelles Lernen* verwendet. Ein Hauptanwendungsgebiet künstlicher Intelligenz ist dabei die Wissensrepräsentation (*knowledge representation*) und darauf aufbauendes symbolisches Schlussfolgern (*symbolic reasoning*), um bspw. Handlungsempfehlungen abzuleiten. Eine gute Übersicht über verschiedene Möglichkeiten zur Wissensrepräsentation für automatisierte Fahrzeuge bietet Pellkofer²⁵⁶, der selbst einen Ansatz zur semantischen Umgebungsbeschreibung mittels linguistischen Variablen verfolgt²⁵⁷ sowie einen darauf aufbauenden Ansatz zur Verhaltensentscheidung mittels einer unscharfen Regelbasis (*fuzzy logic*). Weitere Anwendungsgebiete maschinellen Lernens liegen in der Bildverarbeitung und Objekterkennung²⁵⁸. Eine Schwierigkeit bei Algorithmen des maschinellen Lernens in Bezug auf automatisierte Fahrzeuge ist, dass sie in einer hochdynamischen Umgebung und unter limitierten Ressourcen verfügbar sein müssen und entsprechend nicht über die Leistungsfähigkeit eines sog. *Offline-Expertensystems* verfügen, in dem der Mensch als Experte das Ergebnis mit beeinflusst. Im Kontext der Strukturierung einer funktionalen Architektur kann demnach festgehalten werden, dass wissensbasiertes Verhalten nach der Definition nach Rasmussen kein notwendiger Bestandteil der Architektur sein muss, sondern vielmehr ein möglicher Implementierungsansatz für einzelne Module darstellt.

Werden die Abstraktionsmodelle von Donges und Rasmussen hinsichtlich der Eignung zur Übertragbarkeit auf die hierarchische Strukturierung einer hybriden Architektur gegenübergestellt, so weist das Vorbild nach Donges wesentliche Vorteile auf, da die drei Ebenen in einen Zusammenhang mit den zu implementierenden Funktionen gebracht werden können, die ein automatisiertes Fahrzeugführungssystem besitzen muss. Im Gegensatz dazu muss bei Rasmussen eine weitere Beschreibungsebene vorhanden sein, die die Verhaltensarten auf tatsächliche Systemfunktionen abbildet. Dieser Umstand wirkt sich insbesondere auf die nichtfunktionale Anforderung 13 aus Tabelle 2-2,

²⁵⁵ siehe auch Hertzberg & Chatila (2008): *AI Reasoning Methods for Robotics*, S. 207

²⁵⁶ Pellkofer (2003): *Verhaltensentscheidung für autonome Fahrzeuge...*, S. 34 ff

²⁵⁷ z. B. „Fahrzeug 3 fährt auf Spur 0 und ist langsamer als Fahrzeug 2“, siehe Bspe. in Pellkofer, S. 56



²⁵⁸ Hertzberg & Chatila (2008): *AI Reasoning Methods for Robotics*, S. 208

d.h. die semantisch möglichst einfache und eindeutige Beschreibbarkeit der Module (in diesem Fall der Systemebenen), negativ aus.

Eine gute Zusammenfassung des Vergleichs der beiden Modelle liefert Matthaei^{259a}, wobei er aufzeigt, dass Donges die Aufgaben und Funktionen eines automatisierten Systems beschreibt, wohin gegen Rasmussen die Art und Weise beschreibt, wie Systemfunktionen umgesetzt bzw. implementiert werden können. So lassen sich z. B. Routen- und Trajektorienplanungs-Algorithmen basierend auf einem A*-Algorithmus als regelbasierte Verfahren klassifizieren^{259b}. Gemäß der Definition der funktionalen Systemarchitektur in Kap. 1.2.1 soll diese jedoch gerade unabhängig von konkreten technischen Lösungs- und Implementierungsansätzen sein. Aus diesen Gründen wird das 3-Ebenen Modell nach Donges als Vorbild für die weitere Entwicklung der Referenzarchitektur herangezogen. Im nächsten Abschnitt werden auf dieser Basis die detaillierten Aufgaben und Zuständigkeiten der Systemebenen ermittelt.

²⁵⁹ Matthaei (2015): Wahrnehmungsgestützte Lokalisierung in fahrstreifengenauen Karten.... a: S. 22; b: S. 51

Tabelle 5-2: Semantische Zuordnungstabelle der Abstraktionsebenen (Leserichtung siehe Pfeile)

Ebenen bzw. Aufgaben nach Donges			Ebenen bzw. Verhalten nach Rasmussen				
Navigati-on	Bahn-führung	Stabilisie-rung	Wissensba-siertes Verhal-ten	Regelbasier-tes Verhal-ten	Fertigkeits-basiertes Verhalten		
setzt voraus			Keine eindeutige Zuordnung möglich (hängt von Erfahrung des Operators ab)			Eingabe Navigati-onsziel	Eingriffsmöglichkeiten des Operators
	entspricht der					Vorgabe von Fahr-manövern	
		entspricht der			entspricht in der Regel der	Längs- und Quer-Regelauf-gabe	Automationsgrade
wird umgesetzt in	wird umgesetzt in	wird umge-setzt in	Keine eindeuti-ge Zuordnung möglich	muss imple-mentiert sein in	muss imple-mentiert sein in	Vollauto-matisiert	
	wird umgesetzt in	wird umge-setzt in		muss imple-mentiert sein in	muss imple-mentiert sein in	Teilauto-matisiert	
		wird umge-setzt in			muss imple-mentiert sein in	Assistiert	
setzt Kenntnis voraus über			Keine eindeuti-ge Zuordnung möglich			Straßen-netz	Hierarchische Struktur der Umwelt
	setzt Kenntnis voraus über			setzt Kennt-nis voraus über		Struktur-elemente (Fahrstreifen)	
		setzt Kenntnis voraus über		setzt Kennt-nis voraus über	setzt Kennt-nis voraus über	geometri-sche Be-schreibung der Struk-tur-elemente	

5.2 Kopplung der Ebenen

Im vorliegenden Abschnitt werden die Entscheidungskompetenzen der drei Ebenen sowie die Interaktionen dieser festgelegt. Dazu erfolgt im ersten Schritt eine vertiefende Betrachtung der Informationsverarbeitungsschritte des Modells nach Donges.

5.2.1 Detaillierung der Kontrollhierarchie

Die Drei-Ebenen-Hierarchie der Fahraufgabe nach Donges stellt, wie im vorherigen Kapitel erläutert, ein geeignetes Abstraktionsmodell für die Aufteilung der Systemebenen in hybriden Systemarchitekturen dar. Zur Ableitung der Entscheidungskompetenzen der einzelnen Ebenen sowie der Schnittstellen dieser scheint das Modell jedoch nicht granular genug. Dies wird insbesondere bei näherer Betrachtung der (mittleren) Führungsebene deutlich, die im Gegensatz zu den anderen beiden Ebenen ein enormes Spektrum der funktionalen Anforderungen - das Ermitteln von Führungsgrößen (Sollwerten) für die Regler auf Stabilisierungsebene auf Basis einer vorgegebenen Route - und somit ein Großteil der komplexen Fragestellungen des automatisierten Fahrens adressiert. Aus diesem Grund erfolgt zunächst eine Detaillierung der Kontrollhierarchie unter Betrachtung der Entscheidungen, die das automatisierte Fahrzeugführungssystem (bzw. der Fahrer im teilautomatisierten Betrieb) treffen muss. Das Ergebnis dieses Prozesses ist exemplarisch in Abbildung 5-2 dargestellt und wird im Folgenden näher erläutert. Hierbei sei angemerkt, dass aus Gründen der Übersichtlichkeit zunächst nur auf die deliberativen Aspekte der Hierarchie eingegangen wird und die Betrachtung dynamischer Objekte erst im späteren Verlauf des Kapitels erfolgt.

Ausgangspunkt des Modells nach Donges ist die Transportaufgabe bzw. *Fahrmission*, die es innerhalb des gegebenen Straßennetzes zu bewältigen gilt und als Eingangsinformationen für die *Routenplanung* benötigt wird. Das Routenplanungsmodul bestimmt entsprechend der zur Fahrmission passenden Routensuchkriterien (z. B. kürzeste / schnellste Route, Meide-Kriterien) die optimale Route innerhalb eines als Graph repräsentierten Straßennetzes^{260 261}. Gemäß dem Modell nach Donges gilt damit die Navigationsaufgabe bereits als abgeschlossen (vgl. Abbildung 5-1, rechts).

²⁶⁰ vgl. Kleine-Besten et al (2015): Navigation und Verkehrstelematik, S. 1054

²⁶¹ Ein verbreiteter Standard hinsichtlich eines Datenformats für Navigationskarten stellen die Geographic Data Files (GDF) dar, die in der Norm ISO 14825:2011 (2011): Intelligent transport systems - Geographic Data Files (GDF) - GDF5.0 beschrieben werden.

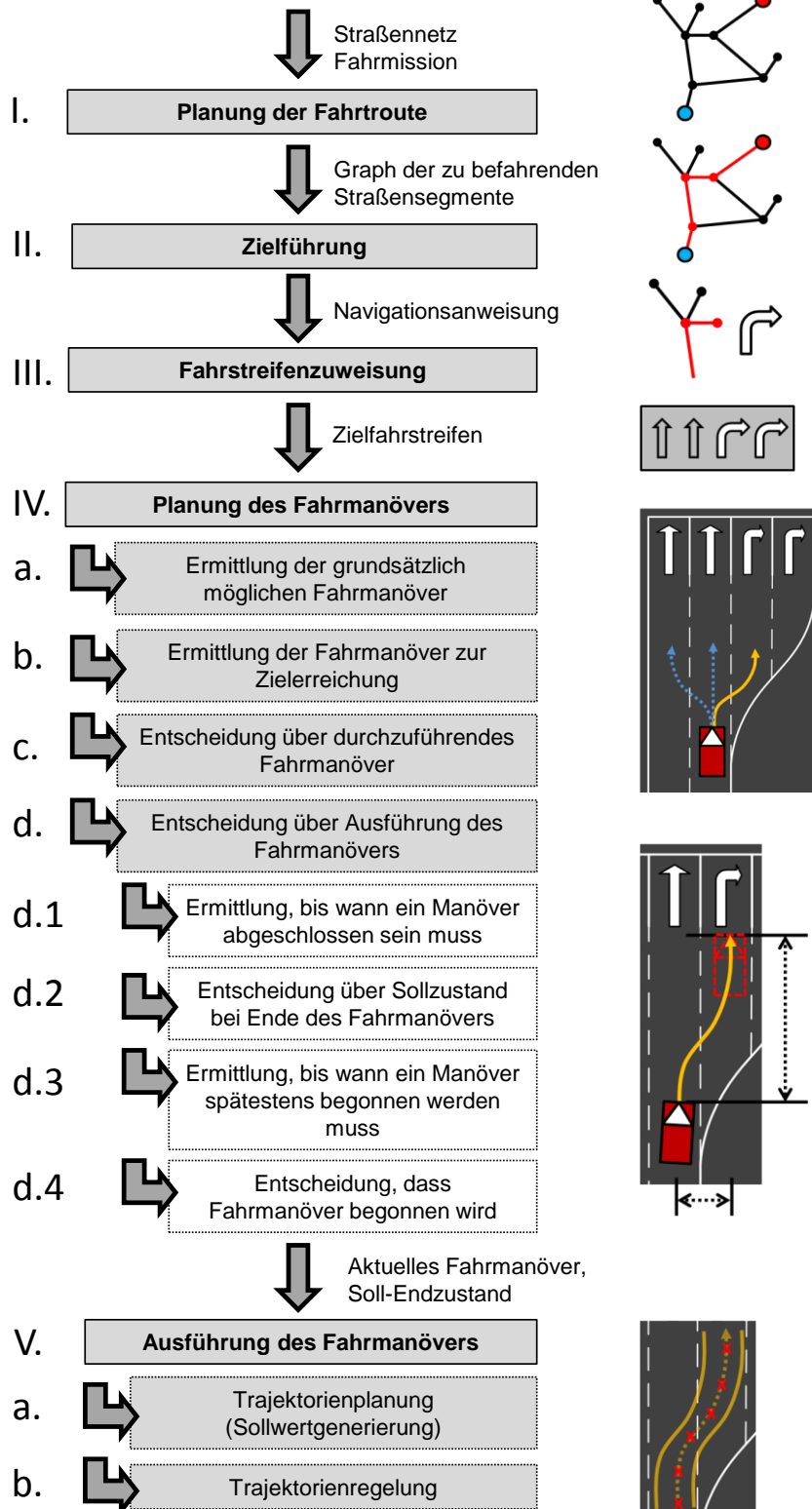


Abbildung 5-2: Detaillierte Kontrollhierarchie auf Basis des Modells von Donges

Um die ausgewählte Route effektiv verfolgen zu können, ist gemäß Kleine-Besten et al.²⁶² jedoch zusätzlich eine *Zielführung* notwendig, wobei dem Fahrer in Serien-Navigationssystemen rechtzeitig vor Erreichen von Entscheidungspunkten visuelle und auditive Hinweise in Form von Navigationsanweisungen gegeben werden. Die Art und Weise der Darreichung der Information einschließlich verwendeter Grammatik, zeitlicher Abfolge und Beschreibungs-Parameter sind dabei herstellerspezifisch und nicht genormt. Hierbei ist festzuhalten, dass die auf dieser Systemebene beschriebenen Navigationsanweisungen nicht mit den in Kap 5.1.2 beschriebenen Fahrmanövern (respektive den Verhaltensweisen in Tabelle 2-1) gleichzusetzen sind und sich auch nicht auf die Strukturelemente (insbes. Fahrstreifen) beziehen, sondern auf den Graph des Straßennetzes. Eine Anweisung „Rechtsabbiegen auf XY-Straße“²⁶³ kann demnach bedeuten, die Verhaltensweise „Fahrstreifenwechsel rechts“ auf einen Abbiegefahrstreifen durchzuführen.

Um diese Informationslücke zu schließen und dem Fahrer möglichst eindeutige Anweisungen darbieten zu können, erfolgt in modernen Navigationssystemen eine *Fahrstreifenzuweisung* sowie eine damit verbundene abstrahierte optische Darstellung der Szene einschließlich der Zielfahrstreifen, die die Umsetzung der Navigationsanweisung ermöglicht. Je nach Datenformat der Navigationskarte ist eine solche Darstellung Teil der Datenstruktur, wie z. B. beim sog. *Junction View Building Block* des Navigation Data Standard Format (NDS)²⁶⁴. Weitere Beispiele für die Fahrstreifenzuweisung sind u.a. in der *Lane Guidance* von Google²⁶⁵ oder im „*Fahrspurassistenten*“ von TomTom²⁶⁶ gegeben. Da im automatisierten Fahrbetrieb jedoch nicht unbedingt (nur) der Fahrer der Empfänger der Navigationsanweisungen und Zielfahrstreifen ist, sondern auch unterlagerte Module der Automationsfunktion, muss als Ausgangsschnittstelle neben einer optischen Darstellung dieser Informationen auch eine Referenz auf die zu erreichende Kante des Kartengraphen respektive auf den Zielfahrstreifen ausgegeben werden, die die unterlagerten Ebenen verarbeiten können.

In der darauf folgenden untergeordneten Hierarchieebene erfolgt die *Planung des Fahrmanövers*, das zur Erreichung der Navigationsanweisungen erforderlich ist (vgl.

²⁶² vgl. Kleine-Besten et al (2015): Navigation und Verkehrstelematik, S. 1049

²⁶³ Zahlreiche Beispiele für Navigationsanweisungen lassen sich leicht durch Abfrage von Online-Routenplanern, generieren, siehe z. B. <https://www.google.de/maps>; <http://www.falk.de/routenplaner>, letzter Zugriff am 03.10.16

²⁶⁴ Behrens et al. (2015): Digitale Karten im Navigation Data Standard Format, S. 520

²⁶⁵ siehe <http://googlepolicyeurope.blogspot.de/2014/12/life-in-fast-lane-lane-guidance-for.html>, letzter Zugriff am 03.10.16

²⁶⁶ siehe http://de.support.tomtom.com/app/answers/detail/a_id/9579/~was-ist-der-fahrspurassistent%3F, letzter Zugriff am 03.10.16

Abbildung 5-2). Dabei findet gemäß Kap. 5.1.2 eine Sequenzierung der Fahrmission in Fahrmissionsabschnitte (Fahrmanöver) statt, wie sie in Tabelle 2-1 dargestellt sind²⁶⁷. Im Kontext der Systemhierarchie ist die Betrachtung dieser in Donges zunächst nicht vorhandenen (Zwischen-) Ebene deswegen zulässig und notwendig, da sie sich - wie in Kap. 2.2.2 gezeigt - auf die Strukturelemente der Fahrstreifen beziehen, deren Nutzung in der Straßenverkehrsordnung eindeutig beschrieben ist²⁶⁸ und darüber hinaus den praktischen Nutzen der besseren Präzifizierbarkeit der Verhaltensweisen der Verkehrsteilnehmer untereinander bietet²⁶⁹, was die Verkehrssicherheit positiv beeinflusst.

Im Allgemeinen lässt sich unter Voraussetzung der Kenntnis der Lage und Anordnung der Fahrstreifen sowie der Art der Fahrstreifenmarkierung zu jedem Zeitpunkt und Ort die Menge der durchführbaren (im Sinne von gemäß StVO erlaubten) Fahrmanöver $M_{durchführbar}$ ermitteln. Am Beispiel in Abbildung 5-2 (Anfahrt einer Kreuzung) wäre demnach auf Basis von Tabelle 2-1 am eingezeichneten Standort des Fahrzeugs

$$M_{durchführbar} = \{FS_{befahren}, FSW_{links}, FSW_{rechts}\}$$

mit den Elementen *Fahrstreifen_befahren* sowie *Fahrstreifenwechsel* mit den Parametern *links* und *rechts* gegeben. Eine Teilmenge davon stellen die Manöver zur Zielerreichung dar, d.h. die Manöver, die zur Erreichung der im vorherigen Absatz beschriebenen Zielfahrstreifen notwendig sind. Bezogen auf das Beispiel in Abbildung 5-2 gilt demnach für die Kreuzungszufahrt

$$M_{Zielerreichung} = \{FSW_{rechts}\}.$$

Unter der Voraussetzung, dass die Automation nur Manöver auswählt, die der Zielerreichung dienen (deliberatives Verhalten) müsste demnach im Rahmen der tatsächlichen Manöverentscheidung auf Basis von in Abbildung 5-2 gelten

$$M_{Auswahl} = \{FSW_{rechts}\}.$$

Im Allgemeinen gilt somit

$$M_{Auswahl} \subseteq M_{Zielerreichung} \subseteq M_{durchführbar},$$

wobei, wie im Beispiel gezeigt, nicht zwangsläufig echte Teilmengen vorliegen müssen.

Für eine derartige Betrachtung ist die Kenntnis über die Lage und Anordnung der Fahrstreifen zueinander von maßgebender Bedeutung, weshalb eine detailliertere Repräsen-

²⁶⁷ Voraussetzung zur Sequenzierung ist hierbei, dass die Fahrmanöver die Fahrmission vollständig abdecken können. Je nach Anwendungsdomäne können hierfür unterschiedliche Manöver notwendig sein (vgl. Tabelle 2-1)

²⁶⁸ vgl. Straßenverkehrsordnung, §7: Benutzung von Fahrstreifen durch Kraftfahrzeuge

²⁶⁹ vgl. Werling (2010): Ein neues Konzept für die Trajektoriengenerierung..., S. 28

tation der Umwelt im Vergleich zur Navigationsaufgabe notwendig ist. Matthaei²⁷⁰ spricht in diesem Zusammenhang von einer *fahrstreifengenauen Weltmodellierung*, in der die lokale Szenerie einschließlich des Szenerie-Kontextes, wie z. B. durch Verkehrszeichen vorgegebene Verkehrsregeln, modelliert und repräsentiert wird (vgl. Abschn. 5.4).

Wurde das grundsätzlich auszuführende Fahrmanöver ausgewählt, so kann im nachfolgenden Schritt über die Ausführung des Manövers entschieden werden, wobei bei expliziten Fahrmanövern (vgl. Kap. 5.1.2) der raumzeitliche Punkt des Beginns sowie die Ausführungsparameter des Manövers, insbesondere hinsichtlich der fahrdynamischen Randbedingungen, ermittelt werden müssen. Die gesuchten Größen lassen sich dabei z. B. über eine Rückwärtspropagierung ausgehend vom zukünftigen Soll-Zustands des Fahrzeugs gewinnen²⁷¹. Die Frage, bis wann ein zur Zielerreichung notwendiges (deliberatives) Manöver abgeschlossen sein muss, lässt sich durch Analyse der Szenerie beantworten. So bestimmt im Beispiel in Abbildung 5-2 das Ende des Abbiegefahrstreifens bzw. der Beginn einer evtl. vorhandenen Sperrlinie die letztmögliche Position des Fahrzeugs beim Endzustand des Fahrtreifenwechsel-Manövers. Darüber hinaus können weitere Sollwerte des Fahrzeugzustands ermittelt werden, wie z. B. die zulässige Längsgeschwindigkeit, die sich aus einem direkt anschließenden Abbiegemanöver über die für den Fahrer zumutbare bzw. gewünschte Querbeschleunigung ergibt. Auf Basis dieses Sollzustands und eines Ausführungsmodells des Manövers lässt sich auf diese Weise ebenfalls der letztmögliche raumzeitliche Punkt des Beginns des Fahrmanövers ermitteln, sodass sich ein strecken- bzw. zeitbasierter Spielraum für die Entscheidung zum tatsächlichen Beginn des Manövers ausgehend vom aktuellen Fahrzeugzustand ergibt.

Die Führungsebene nach Donges umfasst darauf aufbauend gemäß Abbildung 5-1 (rechts) ebenfalls die Ermittlung der „Sollspur“²⁷² und der Sollgeschwindigkeit²⁷³, wobei die beiden Größen gemäß Winner et al.²⁷⁴ in eine Soll-Trajektorie des Fahrzeugs überführt werden können. Auf Basis der soeben beschriebenen Manöverplanung kann unter Berücksichtigung des ermittelten Sollzustandes nach Manöverende sowie der Kenntnis des Zeitpunkts, wann das ausgewählte Manöver begonnen werden soll, eine solche Trajektorie berechnet werden. Die Trajektorie dient im nächsten Schritt als Füh-

²⁷⁰ Matthaei (2015): Wahrnehmungsgestützte Lokalisierung in fahrstreifengenauen Karten..., S. 47 ff

²⁷¹ Lotz & Winner (2014): Maneuver Delegation and Planning for Automated Vehicles...

²⁷² Anmerkung: Es wird vermutet, dass hierbei die Bahn des Fahrzeugs gemeint ist.

²⁷³ als Funktion der Zeit oder der bahnbezogenen zurückgelegten Strecke

²⁷⁴ Winner et al. (2015): Handbuch Fahrerassistenzsysteme, S. 1199

rungsgröße für die unterlagerte Stabilisierungsebene, die die Regelaufgabe der Ansteuerung geeigneter Aktoren zur Beeinflussung der Fahrdynamik ausführt.

Die in diesem Abschnitt erfolgte Detaillierung der Kontrollhierarchie, insbesondere im Hinblick auf Abbildung 5-2, zeigt auf, dass das Modell nach Donges ein Ungleichgewicht der Kontroll- und Entscheidungskompetenzen zwischen den drei Ebenen der Navigations-, (Bahn-) Führungs- und Stabilisierungsebene erzeugt. So wäre lediglich das erste Hierarchieelement (Planung der Fahrroute) der Navigationsebene sowie das letzte Hierarchieelement (Trajektorienregelung) der Stabilisierungsebene und alle weiteren Elemente dazwischen der Führungsebene zuzuordnen.

Mit Blick auf die Checkliste der nichtfunktionalen Anforderungen aus Tabelle 2-2 wird deutlich, dass ein funktional überladenes Führungsebenen-Modul in mehrerlei Hinsicht als ungünstig zu bewerten wäre. Insbesondere die Forderung nach Separation-of-Concerns (Eintrag Nr. 8), die Prüfung auf Modulgröße (Nr. 11) sowie die Prüfung auf semantische Kohäsion (Nr. 13) sind in diesem Zusammenhang zu nennen. Hertzberg et al.²⁷⁵ fassen den Umstand eines Ungleichgewichtes der Modulkompetenzen im Kontext hybrider Architekturen wie folgt zusammen:

„Die Strukturierung eines Problems, die das Problem in drei Teilprobleme zerlegt, von denen eines wiederum deutlich größer ist als die beiden anderen, ist eigentlich keine gute Strukturierung“.

Um dieses Ungleichgewicht zu beseitigen, gibt es prinzipiell zwei Möglichkeiten sowie deren Kombination:

- Kompetenzen der Führungsebene werden teilweise auf die anderen beiden Ebenen verteilt.
- Die Führungsebene wird in mindestens zwei weitere Teilebenen zerlegt.

Im nächsten Abschnitt werden unter Berücksichtigung dieser Optionen verschiedene Varianten der Kontrollhierarchie hergeleitet und bewertet.

5.2.2 Kontrollhierarchie zwischen Navigations- und Führungsebene

Um konkrete Hierarchievarianten erzeugen und diskutieren zu können, werden die Erkenntnisse des letzten Kapitels in Tabelle 5-3 zusammengefasst.

²⁷⁵ Hertzberg et al. (2012): Mobile Roboter, S. 325

Tabelle 5-3: Übersicht über Schnittstellen und Entscheidungsspielräume der einzelnen Hierarchieelemente (Abkürzung: n.v.: nicht vorhanden)

Nr.	Bezeichnung	Eingangsgrößen aus höherer Systemebene	notwendige Eingangsgrößen (Szenerie)	notwendige Eingangsgrößen (Fahrer)	optionale Einflussparameter (Fahrer)	Entscheidungsspielraum	Ausgangsgrößen
I.	Planung der Fahrtroute	n.v.	Straßennetz	Vollautomatisierung: Fahrmission (Ziel)	Suchkriterien (kürzeste / schnellste Route, Meidekriterien)	Auswahl der zu befahrenden Straßen-segmente im Straßennetz	Graph der zu befahrenden Straßen-segmente
II.	Zielführung	Graph der zu befahrenden Straßen-segmente	Straßennetz	n.v.	n.v.	n.v.	Navigationsanweisung
III.	Fahrstreifenzuweisung	Navigationsanweisung	Kenntnis über Fahrstreifen Konstellation (mindestens) an Knotenpunktzufahrt	n.v.	n.v.	n.v.	Referenz auf mögliche Ziel-fahrstreifen
IV.	Planung des Fahrmanövers		Allgemein: Zul. Höchstgeschwindigkeit nach StVO		Allgemein: Wunschgeschwindigkeit des Fahrers		
IV.a	Ermittlung der grundsätzlich möglichen Fahrmanöver	n.v.	Szenerie-Strukturelemente (fahrstreifengenaue Karte, Verkehrszeichen, Lichtsignalanlagen, Fahrbahnmarkierung)	n.v.	n.v.	n.v.	$M_{\text{durchführbar}}$
IV.b	Ermittlung der Fahrmanöver zur Zielerreichung	Referenz auf mögliche Ziel-fahrstreifen, $M_{\text{durchführbar}}$	Fahrstreifengenaue Karte	n.v.	n.v.	n.v.	$M_{\text{Zielerreichung}}$
IV.c	Entscheidung über durchzuführendes Fahrmanöver	$M_{\text{Zielerreichung}}$	Fahrstreifengenaue Karte	Teilautomatisierung : Manöverwunsch des Fahrers (Manöverdelegation)	Vollautomatisierung: Manöverwunsch des Fahrers (Manöverdelegation)	Elemente von $M_{\text{Zielerreichung}}$	M_{auswahl}

Tabelle 5-3: Übersicht über Schnittstellen und Entscheidungsspielräume der einzelnen Hierarchieelemente (Abkürzung: n.v.: nicht vorhanden) (Fortsetzung)

Nr.	Bezeichnung	Eingangsgrößen aus höherer Systemebene	notwendige Eingangsgrößen (Szenerie)	notwendige Eingangsgrößen (Fahrer)	optionale Einflussparameter (Fahrer)	Entscheidungsspielraum	Ausgangsgrößen
IV.d	Entscheidung über Ausführung des Fahrmanövers						
IV.d.1	Ermittlung, bis wann ein Manöver abgeschlossen sein muss	M_{auswahl}	Szenerie-Strukturelemente (fahrstreifengenaue Karte, Verkehrszeichen, Lichtsignalanlagen, Fahrbahnmarkierung)	n.v.	n.v.	n.v.	Raumzeitlicher Punkt, an dem Manöver abgeschlossen werden muss
IV.d.2	Entscheidung über Sollzustand bei Ende des Fahrmanövers	Raumzeitlicher Punkt, an dem Manöver abgeschlossen werden muss	Szenerie-Strukturelemente (fahrstreifengenaue Karte, Verkehrszeichen, Lichtsignalanlagen, Fahrbahnmarkierung)	n.v.	Komfortkriterien (z. B. max. zulässige Quer- und Längsbeschleunigung)	Werte des Sollzustandes im zulässigen Werteintervall der Komfortkriterien und StVO (zul. Höchstgeschwindigkeit)	Soll-Bewegungszustand des Fahrzeugs bei Manöverende
IV.d.3	Ermittlung, bis wann ein Manöver spätestens begonnen werden muss	Soll-Bewegungszustand des Fahrzeugs bei Manöverende	Szenerie-Strukturelemente (fahrstreifengenaue Karte, Verkehrszeichen, Lichtsignalanlagen, Fahrbahnmarkierung)	n.v.	Komfortkriterien (z. B. max. zulässige Quer- und Längsbeschleunigung)	n.v. (da Maximalwerte der Komfortkriterien genutzt werden)	Raumzeitlicher Punkt, an dem Manöver spätestens begonnen werden muss
IV.d.4	Entscheidung, dass Fahrmanöver begonnen wird	Raumzeitlicher Punkt, an dem Manöver spätestens begonnen werden muss	Szenerie-Strukturelemente (fahrstreifengenaue Karte, Verkehrszeichen, Lichtsignalanlagen, Fahrbahnmarkierung), Lichtsignalanlagen-Phase	n.v.	Fahrer fordert Durchführung an bzw. gibt Manöver frei	raumzeitliches Intervall zwischen aktuellem und spätestem Zeitpunkt / Ort des Manöverbeginns	Aktuell durchzuführendes Fahrmanöver, ggf. schematische Handlungen (wie z. B. Aktivierung des Fahrtrichtungsanzeigers)

Tabelle 5-3: Übersicht über Schnittstellen und Entscheidungsspielräume der einzelnen Hierarchieelemente (Abkürzung: n.v.: nicht vorhanden) (Fortsetzung)

Nr.	Bezeichnung	Eingangsgrößen aus höherer Systemebene	notwendige Eingangsgrößen (Szenerie)	notwendige Eingangsgrößen (Fahrer)	optionale Einflussparameter (Fahrer)	Entscheidungsspielraum	Ausgangsgrößen
V.	Ausführung des Fahrmanövers						
V.a	Trajektorienplanung (Sollwertgenerierung)	Aktuell durchzuführendes Fahrmanöver, Soll-Bewegungszustand des Fahrzeugs bei Manöverende, Raumzeitlicher Punkt, an dem Manöver abgeschlossen werden muss	Geometrische Informationen der Szenerie (Fahrstreifenverläufe, Belegt- und Freibereiche)	n.v.	Parametrierung der Trajektorie (z. B. lateraler Versatz)	Verlauf der Trajektorie unter Beachtung des Soll-Endzustandes und -Ortes	Soll-Fahrzeugtrajektorie
V.b	Trajektorienregelung	Soll-Fahrzeugtrajektorie	n.v.	n.v.	n.v.	Konkrete Umsetzung des geforderten Bewegungsvektors mit zur Verfügung stehenden Aktoren	Sollwerte für Fahrzeug-Aktoren

Darin sind die in Abbildung 5-2 beschriebenen (Teil-) Ebenen der Hierarchie aufgeführt, einschließlich der notwendigen Ein- und Ausgangsgrößen, evtl. vorhandener Entscheidungsspielräume sowie dafür relevante Einflussparameter.

Im ersten Schritt wird die Schnittstelle zwischen der Navigations- und Führungsebene und damit einhergehend deren funktionale Kompetenzen festgelegt.

Variantenbildung

Hierarchievarianten können auf Basis von Tabelle 5-3 auf sehr einfache Art und Weise erzeugt werden, indem die Auftrennung der beiden Ebenen durch einen horizontalen Schnitt der Tabellen-Zeilen erfolgt. So ergeben sich die in Tabelle 5-4 aufgeführten Varianten für die Aufteilung der Navigations- und Führungsebene.

Tabelle 5-4: Kontrollhierarchievarianten (Navigations - und Führungsebene); Abkürzungen: V1: Variante 1; NE: Navigationsebene; FE: Führungsebene; &|: und / oder; M: Mensch

Nr.	Bezeichnung	V1	V2	V3	V4	V5	teilautom.
I.	Planung der Fahrtroute	NE	NE	NE	NE	NE	M
II.	Zielführung	FE	NE	NE	NE	NE	M
III.	Fahrstreifenzuweisung	FE	FE	NE	NE	NE	M
IV.	Planung des Fahrmanövers						
IV.a	Ermittlung der grundsätzlich möglichen Fahrmanöver	FE	FE	FE	NE & FE	NE & FE	FE
IV.b	Ermittlung der Fahrmanöver zur Zielerreichung	FE	FE	FE	NE	NE	M
IV.c	Entscheidung über durchzuführendes Fahrmanöver	FE	FE	FE	FE	NE	M
IV.d	Entscheidung über Ausführung des Fahrmanövers	FE	FE	FE	FE	NE & FE	FE

- **Variante 1 (Donges, klassisch)** entspricht hierbei der klassischen Definition nach Donges (vgl. Abschn. 5.1.1 bzw. Abbildung 5-1, rechts), in der die Navigationsebene rein für die Planung der Fahrtroute zuständig ist und der untergeordneten Führungsebene einen Graph der zu befahrenden Straßensegmente übergibt.
- In **Variante 2 (Donges, verschoben auf Zielführung)** wird die Navigationsebene um die Zielführung erweitert, sodass statt eines Graphen eine Navigationsanweisung vor Entscheidungspunkten (wie z. B. Verkehrsknoten) übergeben wird.
- In **Variante 3 (Donges, verschoben auf Fahrstreifenzuweisung)** erfolgt eine Konkretisierung der Navigationsanweisung, sodass bereits eine Referenz auf die zur Anweisung passenden Zielfahrstreifen an die unterlagerte Führungsebene übergeben wird.
- **Variante 4 (Donges, verschoben auf Manöverspielraum)** erweitert die vorherige Variante maßgeblich dadurch, dass bereits eine Menge an zur Zielerreichung möglicher Fahrmanöver ($M_{\text{Zielerreichung}}$) an die Führungsebene übergeben wird.
- Denkbar ist weiterhin eine **Variante 5 (Donges, verschoben auf Manöverauswahl)**, in der die Navigationsebene bereits ein konkretes Fahrmanöver zur Zielerreichung auswählt (M_{Auswahl}).

Im Rahmen dieser Betrachtung steigt der Umfang der funktionalen Kompetenzen der Navigationsebene von Variante 1 bis Variante 5 stetig an. Die Entscheidungskompetenz hingegen bleibt für die Varianten 1-4 gleich, da gemäß Tabelle 5-3 die Hierarchieelemente II., III., sowie IV.a. und IV.b. über keinen eigenen Entscheidungsspielraum verfügen, sondern vielmehr logisch aufeinander aufbauende Informationsverarbeitungsschrit-

te auf Basis des Modells nach Donges darstellen. Variante 5 würde dementsprechend eine Erhöhung der Entscheidungskompetenz darstellen, da die Navigationsebene bereits konkrete Fahrmanöver auswählen könnte.

Prinzipiell ließe sich die Variantenbildung noch weiter fortsetzen. Darauf wird jedoch an dieser Stelle verzichtet, da die Entscheidung zur Ausführung von Fahrmanövern einschließlich der Festlegung von Beginn und Sollzustand des Fahrzeugs bei Ende des Manövers bereits einen starken Einfluss auf die Sollwertgenerierung für unterlagerte Fahrdynamikregler besitzt und somit eindeutig der Führungsebene zuzuordnen ist. Im folgenden Abschnitt werden die beschriebenen Varianten hinsichtlich der Erfüllung der nichtfunktionalen Anforderungen bewertet.

Bewertungskriterien

Um die genannten Varianten bewerten zu können, wird - analog zu den bisherigen Architekturentscheidungen - die Checkliste zur Überprüfung nichtfunktionaler Anforderungen herangezogen (s. Tabelle 2-2). Bei genauerer Betrachtung dieser kann jedoch festgestellt werden, dass sich nicht alle Kriterien unmittelbar zur Bewertung von Architekturentscheidungen auf Modul- / Schnittstellenebene eignen. So ist bspw. Kriterium 6 (Prüfung auf Besitz mehrerer Abstraktionsschichten) inhärent für alle Varianten gegeben. Aus diesem Grund müssen die Kriterien detailliert und um nichtfunktionale Anforderungen, die sich aus den vorherigen Architekturentscheidungen ergeben, erweitert werden. Mit Hinblick auf die Auswahl einer hybriden Schichtenarchitektur ergeben sich die zusätzlichen Anforderungen:

- Unterschiedliche System-Hierarchieebenen sollen sich in einem unterschiedlichen Abstraktionsgrad abbilden (vgl. Kap. 3.3), d.h.:
 - Höhere Ebenen sollen einen größeren raumzeitlichen Horizont überblicken als tiefere.
 - Die Entscheidungsperiode auf einer höheren Ebene soll länger sein als die auf der darunter liegenden.
- Wie bereits in Abschn. 5.2.1 festgestellt wurde, sollten die Ebenen einen ähnlichen Umfang der Modulkompetenzen und kein deutliches Ungleichgewicht besitzen (s. auch Tabelle 2-2, Kriterium Nr. 8, 11, 13)

Um der besonderen Anforderung dieser Arbeit Rechnung zu tragen, mit der zu entwickelnden Architektur verschiedene Automationsgrade abdecken zu können, so wird zusätzlich ein Qualitätsattributsszenario (vgl. Kap. 1.2.4) - im nachfolgenden mit *QAS* abgekürzt - formuliert, das explizit die Anforderung nach Änderbarkeit / Erweiterbarkeit adressiert (vgl. Kap. 2.3.2):

- *Ausgangssituation:* Gegeben sei eine System, dessen funktionale Anforderungen die manöverbasierte Teilautomation vollständig abdeckt, wie sie in Kap. 2.1 für

das Ausführungsbeispiel PRORETA 3 beschrieben ist (vgl. Tabelle 5-4, Spalte „teilautom.“)

- *Quelle / Stimulus:* Das System soll nun vom Entwicklerteam um eine Vollautomatisierung erweitert werden, d.h. im speziellen, dass der Fahrer nicht mehr die Manöver an das System delegiert, sondern nur noch das Fahrziel bestimmt.
- *Antwort / erwartetes Ergebnis:* Die bestehenden Modulkompetenzen und Schnittstellen sollen möglichst erhalten bleiben, sodass der Änderungsaufwand möglichst klein ist. Die Änderung sollte ohne ein komplexes Verständnis des bisherigen Systems möglich sein (Forderung nach loser Kopplung, vgl. Tabelle 2-2, Nr. 5).

Variantenbewertung

Variante 1: Donges (klassisch)

Übergibt die Navigationsebene (NE) in Variante 1 die geplante Route als Graph der zu befahrenden Straßenelemente an die Führungsebene (FE), so müsste letztere ebenfalls Zugriff auf die straßengenaue, topologische Karte besitzen, um Navigationsanweisungen an Knotenpunkten ableiten zu können. Letzteres würde dazu führen, dass beiden Ebenen aufgrund der gemeinsamen Datenbasis denselben räumlichen Horizont überblicken. Darüber hinaus würden in der FE unterschiedliche Laufzeiten vorherrschen, da Navigationsanweisungen nur an Knotenpunkten auftreten, Manöverentscheidungen jedoch auch zwischen diesen auftreten können. Hinsichtlich des Umfangs der Modulkompetenzen schafft die Aufteilung in Variante 1 keine Entlastung der FE. Dies kann auch dadurch veranschaulicht werden, dass im Falle des oben geschilderten QAS die FE im Falle der Teilautomation nicht nur für die Ermittlung der grundsätzlich möglichen Fahrmanöver (vgl. Tabelle 5-3, Nr. IV.a.) sowie der Entscheidung über raumzeitliche Ausführung des Fahrmanövers (vgl. Tabelle 5-3, Nr. IV.d) zuständig wäre, sondern darüber hinaus auch das vorher vom Fahrer delegierten Manöver zur Zielerreichung auf Basis der Navigationsanweisung und Fahrstreifenzuweisung selbst bestimmen müsste. Diese Inhalte müssten dem Modul im Falle des QAS hinzugefügt werden, wodurch die bestehende Eingangsschnittstelle, also das gewünschte Manöver des Fahrers, entfallen und durch den Graphen der geplanten Route ersetzt werden müsste. Eine solche Änderung würde ein tieferes Verständnis des FE-Moduls voraussetzen.

Variante 2: Donges, verschoben auf Zielführung

Wird die NE im Vergleich zu Variante 1 um die Zielführung erweitert und übergibt somit eine Navigationsanweisung an Entscheidungs- bzw. Knotenpunkten an die FE, so müsste letztere um einen Algorithmus zum Abgleich („Matching“) der Navigationsanweisung und der fahrstreifengenaue Karte erweitert werden. Dies wäre insbesondere dann der Fall, wenn die straßen- und fahrstreifengenaue Karte getrennte Instanzen

sind²⁷⁶ und nicht, wie in Kap. 5.1.2 beschrieben, in Form eines konsistenten, hierarchisch aufgebauten Datensatzes vorliegen. Hinsichtlich des räumlichen Horizonts wären die beiden Ebenen weitestgehend getrennt, da die FE nicht die gesamte Strecke, sondern nur maximal die bis zum nächsten Entscheidungspunkt berücksichtigen müsste. Dennoch würden in der FE unterschiedliche Laufzeiten vorherrschen, da die Schritte der Fahrstreifenzuweisung sowie eine Entscheidung über die passenden Manöver zur Zielerreichung nur an Entscheidungspunkten durchgeführt werden müssten. Im Falle des QAS wäre die FE ähnlich wie in Variante 1 deutlich zu erweitern, wobei die Schnittstelle zwischen NE und FE in Variante 2 durch die einfachere Beschreibbarkeit einer Navigationsanweisung statt eines Graphen weniger komplex wäre.

Variante 3: Donges, verschoben auf Fahrstreifenzuweisung

In Variante 3 beinhaltet die NE im Vergleich zur Variante 2 zusätzlich die Fahrstreifenzuweisung, was sich insbesondere dann positiv auf die funktionalen Zuständigkeiten der Module auswirken würde, wenn straßen- und fahrstreifengenaue Karte auf der gleichen, hierarchisch aufgebauten Datenbasis beruhen, da als Schnittstelle zwischen den Ebenen neben der Navigationsanweisung lediglich eine Referenz auf den / die Zielfahrstreifen an die FE übergeben werden könnte. Ebenfalls positiv hervorzuheben wäre die damit mögliche stringente Trennung der Hierarchieebenen hinsichtlich des überblickten räumlichen Horizonts, da die NE einzig auf die straßengenaue und die FE rein auf die fahrstreifengenaue Karte zugreifen müssten. Hinsichtlich des QAS wäre in der FE die Funktion der Ermittlung der Manöver zur Zielerreichung hinzuzufügen, wobei diese implementierungsseitig ähnlich zu integrieren wäre, wie die ohnehin bereits in der FE notwendige Funktion der Ermittlung der grundsätzlich möglichen Fahrmanöver im Falle der Teilautomation nach Vorbild von PRORETA 3. Konkret würde dies bedeuten, dass in der FE die Menge der möglichen Fahrmanöver $M_{durchführbar}$ nach der Teilmenge der Manöver zur Zielerreichung $M_{Zielerreichung}$ durchsucht werden müsste, die sich aus der Erreichung der übergebenen Zielfahrstreifen und der dazugehörigen Navigationsanweisung ergeben.

Variante 4: Donges, verschoben auf Manöverspielraum

Variante 4 unterscheidet sich dadurch von Variante 3, dass die NE bereits die Manöver zur Zielerreichung $M_{Zielerreichung}$ an Entscheidungspunkten an die FE übergibt. Da sich die möglichen Manöver im Sinne der in Kap. 2.3.3 beschriebenen Verhaltensweisen jedoch aus der relativen Lage des Fahrzeugs auf der fahrstreifengenaue Karte ergeben, müsste die NE Zugriff auf diese sowie eine Aussage zur Lage des Fahrzeugs darauf besitzen. In Bezug auf die obig definierten Anforderungen hinsichtlich des räumlichen

²⁷⁶ Dies wäre bspw. dann der Fall, wenn die Navigationskarte und die fahrstreifengenaue Karte nicht vom gleichen Kartenprovider stammen.

Abstraktionsgrades sowie der Entscheidungsperiode wäre dies negativ hervorzuheben. Im Falle des QAS ergäben sich zwei Optionen, Variante 4 umzusetzen:

1. Ebene IV.a (Ermittlung von $M_{durchführbar}$) verbleibt in der FE, Ebene IV.b (Ermittlung von $M_{Zielerreichung}$) erfolgt unabhängig davon durch eine eigene Logik in der NE.
2. Ebene IV.a wird ebenfalls in die NE verschoben.

Bei Option 1 wäre durch zusätzliche Maßnahmen sicherzustellen, dass die beiden Datensätze zeitlich konsistent sind. Eine solche Maßnahme wäre z. B., dass die FE $M_{durchführbar}$ an die NE übergibt, was die Schnittstellen zwischen den Ebenen vergrößern würde. Option 2 würde im Falle des QAS eine starke Umstrukturierung der Architektur bedeuten, da Teile des Quelltextes aus der FE ausgelöst und in das Modul der NE übertragen werden müssten. Ein komplexes Verständnis der beiden Module wäre Voraussetzung für die beteiligten Entwickler und widerspricht der Forderung nach loser Modul-Kopplung. Ein weiterer Nachteil wäre dann gegeben, wenn im Falle der Vollautomatisierung als Mehrwertfunktion für den Fahrer immer noch die Möglichkeit gegeben werden soll, Fahrmanöver analog zur manöverbasierten Teilautomatisierung eingeben zu können (Vollautomatisierung mit optionaler Manöverdelegation). In diesem Falle müsste die MMS zur Übertragung möglicher Fahrmanöver an den Fahrer von der FE zur NE verschoben werden. Positiv hervorzuheben wäre an Variante 4 jedoch der Aspekt, dass die FE zu Lasten der NE funktional entlastet würde.

Variante 5: Donges, verschoben auf Manöverauswahl

In Variante 5 entscheidet die NE vollumfänglich über das Soll-Fahrmanöver des Fahrzeugs. Für die Bewertung sind die in Variante 4 genannten Aspekte ebenfalls gültig. Darüber hinaus muss an dieser Stelle jedoch eine Besonderheit berücksichtigt werden, die zuvor keine Relevanz besaß: Die Entscheidung über die Auswahl des Fahrmanövers wird zwar in der Regel vor dem Hintergrund der Zielerreichung getroffen, diese kann jedoch Freiheitsgrade hinsichtlich der Auswahl des tatsächlichen Fahrmanövers zulassen, wenn die Anzahl der Elemente in $M_{Zielerreichung}$ größer als 1 ist. Ein Beispiel hierfür wäre das Befahren einer Autobahn, bei dem der Fahrer den Fahrstreifen frei wählen kann, also das Verlassen der Autobahn nicht unmittelbar bevorsteht. Als Einflussgrößen, die in einer solchen Situation auf die Fahrstreifen- und dementsprechend auf die Manöverauswahl einwirken, werden im Folgenden unter dem Begriff *kooperative / taktische Verhaltensweisen* zusammengefasst.

Der Begriff *kooperatives Verhalten* wurde bereits in Kap. 2.2.2 beschrieben. *Taktisches Verhalten* wird im Rahmen dieser Arbeit als ein Verhalten definiert, das (nur) dem eigenen Fahrer-Fahrzeug-System einen Vorteil hinsichtlich des Erfüllungsgrades der Fahr-

mission verspricht, wie bspw. der schnelleren Zielerreichung²⁷⁷. Würde die NE über das tatsächlich auszuführende Fahrmanöver entscheiden, so müsste diese nicht nur Zugriff auf die fahstreifengenaue Karte sowie der Lage des eigenen Fahrzeugs darin besitzen, sondern darüber hinaus die Möglichkeit besitzen, den semantischen Kontext der Verkehrssituation zu verstehen, in dem sich das eigene Fahrzeug befindet. Dies beinhaltet u.a. auch die Kenntnis der aktuellen und zukünftigen Lage anderer Verkehrsobjekte sowie geltender Verkehrs- und Verhaltensregeln, welche zur Ableitung der Notwendigkeit und Nützlichkeit kooperativer / taktischer Verhaltensweisen vorausgesetzt werden. Darüber hinaus und im Gegensatz zu rein deliberativ motivierten Manövern zur Zielerreichung ist bei der Planung kooperativer / taktischer Manöver die Ebene IV.c (Entscheidung über durchzuführendes Manöver) stark mit der Ebene IV.d (Entscheidung über die Ausführung des Fahrmanövers) gekoppelt, da sich aus dem Situationskontext bereits Rahmenbedingungen hinsichtlich des raumzeitlichen „Fensters“ für die Manöverdurchführung ergeben²⁷⁸. Vor diesem Hintergrund würde Variante 5 die Anordnung der Hierarchieelemente in Tabelle 5-3 stark durchmischen, was auch hinsichtlich des überblickten räumlichen Horizonts und vorherrschenden Entscheidungsperioden der Ebenen als nachteilig zu bewerten wäre.

Im Falle des QAS wäre jedoch der Aspekt positiv hervorzuheben, dass die ursprüngliche Architektur der FE nicht geändert werden müsste, da die Ausgangsschnittstelle (Ausgabe der möglichen Fahrmanöver) und die Eingangsschnittstelle (Ausgewähltes Fahrmanöver) statt vom menschlichen Fahrer nun vollständig von der NE bedient würde (vgl. Tabelle 5-4, Spalten „V5“ und „teilautom.“).

Variantenauswahl

Hinsichtlich der durchgeführten Bewertungen lassen sich Zielkonflikte bzgl. der Erfüllung der nichtfunktionalen Anforderungen erkennen: Wird das Kriterium der Abstraktionsgrade der Ebenen nach unterschiedlichen raumzeitlichen Horizonten bzw. Entscheidungsperioden herangezogen, so ergibt Variante 3 die deutlichste Trennung. Für den Fall des Qualitätsattributsszenarios wäre jedoch, wie im vorherigen Abschnitt beschrieben, Variante 5 zu bevorzugen, was bedeutet, dass die Variantenauswahl letztlich von der Gewichtung der nichtfunktionalen Anforderungen abhängt.

²⁷⁷ Ein Beispiel für ein taktisches Verhalten wäre ein Fahstreifenwechsel-Manöver vom rechten auf den linken Fahstreifen einer zweistreifigen Autobahn aufgrund des Annäherns des eigenen Fahrzeugs an eine langsamer fahrende Lkw-Kolonnen.

²⁷⁸ Bsp.: Eigener Fahstreifenwechsel, um einem anderen Fahrzeug die Einfahrt auf den eigenen Fahstreifen zu ermöglichen (Einfädelungstreifen)

Im Rahmen der vorliegenden Arbeit wird Variante 3 zur Trennung der Navigations- und Führungsebene präferiert. Dies liegt, neben den bereits obig erwähnten Vorteilen, insbesondere daran, dass die Navigationsebene keine Informationen über die fahstreifengenaue Lage des eigenen Fahrzeugs sowie anderer Verkehrsobjekte benötigt, um ihre Funktion zu erfüllen. Somit sind sämtliche dynamischen Aspekte des Umfeld- bzw. Weltmodells für sie nicht relevant. Auch wenn die Schnittstelle eine Referenz (z. B. implementierungsspezifische Identifikationsnummern) auf die Zielfahstreifen des nächsten Entscheidungs- bzw. Knotenpunktes an die Führungsebene übergibt, so ist eine genaue Kenntnis des fahstreifengenauen Weltmodells für die Navigationsebene nicht zwingend erforderlich, was insbesondere daran belegt werden kann, dass handelsübliche Navigationssysteme bereits über eine solche Funktion verfügen, ohne dass eine fahstreifengenaue Karte hinterlegt sein muss (s. Abschn. 5.2.1). Eine solche Hierarchieaufteilung schafft für die unterlagerte Führungsebene einen klar definierten Entscheidungsspielraum, die prinzipiell frei über die durchzuführenden Manöver entscheiden kann, solange nicht die Erreichung der Zielfahstreifen (und. ggf. zusätzlich notwendiger Navigationsanweisung) gefährdet ist. Variante 3 schafft somit eine semantisch einfach zu beschreibende Schnittstelle, ermöglicht eine klare Trennung der Modulfunktionen zwischen Navigations- und Führungsebene und erfüllt somit die Forderung nach hoher semantischer Kohäsion sowie der Separation of Concerns (vgl. Kriterien 8 und 13 in Tabelle 2-2).

Eine Aufteilung gemäß Variante 4 und 5 würde zwar die Führungsebene funktional noch weiter entlasten, jedoch zu Lasten der Beschreibbarkeit der funktionalen Zuständigkeiten der Ebenen. Insbesondere Variante 5, die hinsichtlich des QAS der Erweiterbarkeit der Teilautomation eine reizvolle Option darstellt, verwischt die Grundsätze hierarchisch aufgebauter, hybrider Basisarchitekturen, da bereits in der Navigationsebene Informationen zu anderen dynamischen Verkehrsobjekten benötigt würden und somit ein Teil der reaktiven Elemente eines solchen Systems beinhalten würde. Ebenfalls widersprechen Varianten 4 und 5 deutlich dem Abstraktionsmodell von Donges.

5.2.3 Kontrollhierarchie zwischen Führungs- und Stabilisierungsebene

Entsprechend der angewandten Methodik wird im Folgenden die Kompetenzverteilung sowie die Schnittstelle zwischen der Führungs- und Stabilisierungsebene (SE) betrachtet. Gemäß der Definition von Donges besteht die Aufgabe der SE in der (kompensatorischen) Regelaufgabe der in der Führungsebene generierten Führungsgrößen (vgl. Abschn. 5.1.1 sowie Abbildung 5-1) auf Fahrdynamikebene. Gemäß Tabelle 5-3 bzw. Abbildung 5-2 entspräche dies einer Auftrennung der Ebenen V.a und V.b.

Die Generierung und Bewertung von Entscheidungsvarianten wird an dieser Stelle jedoch aufgrund der folgenden Überlegung nur verkürzt dargestellt:

- Wird die Schnittstelle zwischen FE und SE gemäß Tabelle 5-3 nach oben verschoben, sodass die SE neben der Regelaufgabe auch die Trajektorienplanung beinhaltet, so müsste die SE neben dem Zugriff auf interne Systemzustände (wie z. B. die Fahrzeug-Eigendynamik und -Pose) ebenfalls Zugriff auf die Teile des Weltmodells besitzen, die auch die (externe) Verkehrssituation beschreiben. Hinsichtlich der Berücksichtigung der nichtfunktionalen Anforderungen aus Tabelle 2-2 birgt eine solche Aufteilung Nachteile hinsichtlich der Forderung nach Dekomposition (Nr. 3, Aufteilung in Entscheidung / Planung und Ausführung), Modulgröße (Nr. 11) sowie insbesondere hinsichtlich der semantischen Kohäsion (Nr. 13), da beide Teilaufgaben semantisch unabhängig voneinander beschreibbar sind.
- Eine enge Kopplung von Trajektorienplanung und -Regelung ist gemäß Werling²⁷⁹ aus regelungstechnischer Sicht nicht zwingend erforderlich und benötigt nur in seltenen Fällen eine Interaktion in Form einer Rückkopplung (bei impulsförmigen Störungen der Regelstrecke, die eine Reinitialisierung der Trajektorie notwendig macht).
- Wird die Schnittstelle zwischen FE und SE weiter „nach unten“ verschoben (nicht mehr in Tabelle 5-3 dargestellt), sodass bspw. die FE die Stellgrößen für die Aktoren bereitstellt, so ergeben sich signifikante Nachteile hinsichtlich der nichtfunktionalen Anforderungen (z. B. Kopplung der Module mit Hardwarearchitektur und Fahrzeugplattform, Berücksichtigung / Arbitrierung von Fahrdynamik-Stabilisierungsfunktionen etc.). Für eine ausführliche Diskussion und Bewertung dieser Architekturvariante sei an dieser Stelle auf Behere & Törngren²⁸⁰ verwiesen.

Aus den genannten Gründen wird im Rahmen dieser Arbeit eine Aufteilung der beiden Ebenen gemäß der ursprünglichen Definition von Donges präferiert, was einer Teilung gemäß den Zeilen V.a und V.b auf Basis von Tabelle 5-3 entspricht. Im nächsten Abschnitt wird die Führungsebene aufgrund ihrer übergeordneten Bedeutung näher betrachtet.

²⁷⁹ Werling (2010): Ein neues Konzept für die Trajektoriengenerierung..., S. 11 ff

²⁸⁰ Behere & Törngren (2015): A Functional Architecture for Autonomous Driving

5.2.4 Kontrollhierarchie innerhalb der Führungsebene

Aufgabe der Führungsebene

Die Aufgabe der Führungsebene lässt sich folglich definieren als eine Berechnung der Soll-Fahrzeugtrajektorie, die die Erreichung der von der Navigationsebene übergebenen Zielfahrstreifen unter Berücksichtigung der Navigationsanweisung sicherstellen soll. Die Stabilisierungsebene führt anschließend die Regelung der sich daraus ergebenden fahrdynamischen Führungsgrößen aus. Dies impliziert, dass in der Führungsebene gemäß den Grundsätzen einer hybriden Basisarchitektur die deliberativen und reaktiven Anteile des Gesamtsystems kombiniert werden müssen. Im vorliegenden Abschnitt wird deshalb diskutiert, ob eine weitere Aufteilung der Führungsebene in zwei oder mehr Teilelemente zur Modularisierung des Systems beiträgt und wie diese Aufteilung ggf. ausgeführt werden sollte.

Wie bereits in Kap. 3.3 beschrieben, dienen deliberative Anteile des Systems dazu, längerfristige Ziele zu erreichen und die Fahrmission umzusetzen. Die Vorgabe von zu erreichenden Zielfahrstreifen auf Basis der Fahrtroute ist somit zweifelsfrei als eine deliberative Aufgabe zu klassifizieren, ebenso wie bspw. das damit verbundene Durchführen eines Fahrstreifenwechsel- oder Abbiegemanövers. Reaktive Elemente dienen dazu, auf dynamische und ggf. unvorhergesehene Umwelteinflüsse situativ richtig zu reagieren. Somit kann das (Not-) Bremsen auf ein plötzlich erscheinendes Hindernis als ein reaktives Verhalten klassifiziert werden, aber jedoch auch das kollisionsfreie Zusammenspiel mit anderen Verkehrsobjekten wie bspw. ein sicheres Fahrzeugfolgeverhalten.

Im Kontext des automatisierten Fahrens lassen sich jedoch nicht alle Verhaltensentscheidungen als rein deliberativ oder rein reaktiv klassifizieren, was sich insbesondere an kooperativen / taktischen Verhaltensweisen zeigt. So besitzt bspw. das Szenario eines eigenen Fahrstreifenwechsels, um ein langsamer fahrendes Fahrzeug zu passieren, sowohl deliberative (schnellere Erfüllung der Fahrmission) als auch reaktive Anteile (das „einander aus dem Weg gehen“).

Eine weitere Herausforderung der Kopplung der beiden Elemente besteht in der starken Strukturierung und Reglementierung der Umgebung automatisierter Straßenfahrzeuge. Diese resultiert darin, dass dem System diskrete Entscheidungen sowohl auf Navigations- als auch auf Manöverebene aufgezwungen werden²⁸¹, die im Rahmen der Hierarchie jedoch in einen kontinuierlichen Zustandsraum der fahrdynamischen Führungsgrößen

²⁸¹ Es sind nur wenige Verhaltensweisen im Sinne von Fahrmanövern zulässig, die in Ablauf und Durchführungs-Reihenfolge gleichzeitig in den Szenerie-Kontext passen, den im Straßenverkehr geltenden Regeln und Gesetzen entsprechen und, als Sequenz angeordnet, die Zielerreichung sicherstellen.

Ben transformiert werden müssen. Da klassisch reaktive Verhaltensweisen häufig einen kontinuierlichen Zustandsraum voraussetzen (wie z. B. der von Arkin²⁸² propagierte Potenzialfeldansatz), ergeben sich in diesem Kontext entweder starke Einschränkungen bzgl. deren zulässigen Anwendungsgebietes (z. B. reaktive Querführung, aber nur innerhalb eines Fahrstreifens) oder müssen die Möglichkeit besitzen, Einfluss auf diskrete Manöverentscheidungen auszuüben. Bei letzter Option muss dabei zusätzlich sichergestellt werden, dass diese nicht im Widerspruch zu den (deliberativen) Missionszielen stehen.

Vor diesem Hintergrund stellt sich für die funktionale Architektur die Frage, ob sich diese komplexen Wechselwirkungen mit der häufig in hybriden Architekturkonzepten zu findenden Aufteilung zwischen einer diskreten und regelbasierten Vorgabe bzw. Sequenzierung von Verhaltensweisen²⁸³ und einer reaktiven Ausführung dieser abbilden lassen (vgl. Kap. 3.3). Um diese Fragestellung zu untersuchen, werden im folgenden Abschnitt zwei Architekturvarianten untersucht. In der ersten Variante wird versucht, die erwähnte typische Aufteilung zwischen deliberativer Verhaltensentscheidung und reaktiver Ausführung umzusetzen, die zweite Variante stellt ein Repräsentant einer Anordnung dar, die die beiden Elemente enger miteinander verknüpft.

Variantenbildung

Analog zum Vorgehen im vorherigen Abschnitt werden Varianten zunächst durch Variation möglicher Schnittstellen auf Basis von Tabelle 5-3 erstellt. Da die Ebenen IV.a (Ermittlung der grundsätzlich möglichen Fahrmanöver) sowie IV.b (Ermittlung der Fahrmanöver zur Zielerreichung) implizit durch die aktuelle Fahrsituation und -mission vorgegeben sind, besitzen diese keinen eigenen Entscheidungsspielraum und stellen deswegen reine Informationsverarbeitungsschritte dar. Erst in Ebene IV.c (Entscheidung über durchzuführendes Fahrmanöver) besteht ein tatsächlicher Entscheidungsspielraum, das aktuell durchzuführende Fahrmanöver aus der Menge der zulässigen Manöver zur Zielerreichung auszuwählen. Da das Einziehen einer Hierarchieebene ohne eigenen Entscheidungsspielraum gegen die Grundsätze einer hybriden Basisarchitektur ver-

²⁸² Arkin (2000): Behavior-Based Robotics, S. 141 ff

²⁸³ Arkin nennt diese Art der Kopplung von deliberativen und reaktiven Elementen „Selektion“: Die deliberative Komponente bestimmt, welche Verhaltenselemente und deren Parameter zur Laufzeit ausgeführt werden und hat die Befugnis, diese umzukonfigurieren. Siehe: Arkin (2000): Behavior-Based Robotics, S. 214

stößt²⁸⁴, ergibt sich die erste Variante aus einer Auftrennung der Führungsebene durch die Ebenen IV.c und IV.d (Entscheidung über Ausführung des Fahrmanövers). Dies führt zu einer Zweiteilung der Führungsebene in eine obere Ebene, die im Folgenden zunächst als Führungsebene I (FE I) und eine untere Ebene, die als Führungsebene II (FE II) bezeichnet wird.

Die Schwierigkeit, überhaupt Architekturvarianten für die Führungsebene zu definieren, besteht darin, eine Aufteilung zu finden, die über alle (Verkehrs-) Situationen, die zur Laufzeit auftreten können, möglichst konsistent und semantisch eindeutig beschreibbar bleibt (vgl. auch die Forderung nach hoher semantischer Kohäsion in Tabelle 2-2). Dies ist die Voraussetzung dafür, dass eine Architektur im praktischen Einsatz nicht durch „Balkone“ oder ungeplante Schnittstellen („Shortcuts“) unterhöhlt wird. Aus diesem Grund müssen die erzeugten Entscheidungsvarianten hinsichtlich ihrer Übertragbarkeit auf verschiedenste Verkehrssituationen bewertet werden. Zum besseren Verständnis wird zunächst Variante 1 dieser Bewertung unterzogen.

Verkehrssituationen, die dabei betrachtet werden, leiten sich aus der bereits in Kap. 2.2.3 vorgestellten Übersicht der notwendigen Verhaltensweisen für automatisiertes Fahren ab (s. Tabelle 2-1) und werden im Folgenden als Referenzsituationen bezeichnet. Dabei ist von besonderem Interesse, wie die Architekturvariante die notwendigen deliberativen und reaktiven Systemanteile abbildet und ggf. Widersprüche zwischen diesen auflöst. Die Referenzsituationen sind:

1. Fahrstreifen befahren mit vorausfahrendem Fahrzeug: Hierbei ist von Interesse, wie das (reaktive) Abstands- und Folgeverhalten abgebildet wird.
2. Fahrstreifenwechsel zulassen: Es wird diskutiert, welche Systemebene das kooperative Einfädelnlassen anderer Fahrzeuge abdeckt.
3. Annähern an Knoten / Fußgängerüberweg: Wie wird das Annäherungsverhalten an Entscheidungspunkte in der Längsführung umgesetzt?
4. Kooperativer / taktischer Fahrstreifenwechsel in Lücke in Nachbar-Fahrstreifen. Hierbei wird diskutiert, wie das System die Nützlichkeit bzw. Notwendigkeit eines Fahrstreifenwechsels in eine Lücke im Nachbar-Fahrstreifen erkennt und umsetzt.
5. Kooperativer / taktischer Fahrstreifenwechsel bei Annäherung an Knotenpunkt: Hierbei wird untersucht, wie das System einen Fahrstreifenwechsel verhindert,

²⁸⁴ Systeme mit organisatorischer Hierarchie: „Die daraus entstehenden Entscheidungshierarchien zeichnen sich dadurch aus, dass die obere Entscheidungsinstanz die unteren Entscheidungsinstanzen zwar koordiniert, ihnen aber Handlungsspielraum lässt. Die unteren Entscheidungsinstanzen können sich in einem von der oberen Entscheidungsinstanz vorgegebenen Rahmen beeinflussen“, aus Maurer (2000): Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen, S. 33

der im Widerspruch zu den Missionszielen steht (z. B. das Verhindern eines Fahrstreifenwechsels nach links kurz vor einer zu erreichenden Ausfahrt²⁸⁵).

6. Annähern an Kreuzung: Hierbei wird untersucht, wie das System das Erreichen eines Abbiege-Fahrstreifens bzw. Einordnen in einen Fahrstreifen (ohne Interaktion mit anderen Verkehrsobjekten) umsetzt²⁸⁶.
7. Vorfahrtgewähren beim Abbiegen / Kreuzen / bei Einfahrt in Kreisverkehr: Die Fragestellung, welche Architektur-Ebene für das Beachten der Vorfahrt zuständig ist, wird durch dieses Referenzszenario diskutiert.

Tabelle 5-5 (linke Seite) führt die Merkmale von Variante 1 auf. Demnach entscheidet die FE I über das durchzuführende Fahrmanöver, das zur Zielerreichung notwendig ist, nicht jedoch über dessen Ausführungsverlauf einschließlich Beginn und Ende. Dementsprechend müsste die FE II eine dazu passende, kollisionsfreie Trajektorie berechnen. Hieran zeigt sich, dass Variante 1 versucht, die deliberativen Elemente in FE I von den reaktiven Elementen in FE II zu trennen. Das Zusammenspiel der Ebenen wird im Folgenden auf Basis der aufgeführten Referenzszenarien detailliert. Das Ergebnis dieses Schrittes ist in Tabelle 5-6 aufgeführt.

Tabelle 5-5: Varianten zur Aufteilung der Führungsebene

	V1	V2
Bezeichnung	Deliberative Manöverplanung, reaktive Trajektorienplanung	Deliberative, koop./takt. Manöverplanung, reaktive Trajektorienplanung
Aufgabe FE I	FE I entscheidet über durchzuführendes Fahrmanöver, das zur Zielerreichung notwendig ist (deliberativ), nicht jedoch über dessen Ausführungsverlauf.	FE I entscheidet über durchzuführendes Fahrmanöver sowie dessen Ausführungsverlauf. Dabei gibt FE I den zulässigen Entscheidungsspielraum für Trajektorienplanung in FE II explizit vor. Im Unterschied zu V1 entscheidet somit FE I über den tatsächlichen Beginn des Manövers und muss somit bereits eine Kollisionsfreiheit des zukünftigen Verhaltens berücksichtigen.
Aufgabe FE II	Manöver sind in FE II als (reaktive) Verhalten implementiert. FE II entscheidet somit selbst über Beginn und Umsetzung des Manövers sowie den System-Endzustand bei Manöverende. FE II berechnet somit eine zur Vorgabe von FE I passende, kollisionsfreie Trajektorie.	Manöver sind in FE II (analog zu V1) als reaktive Verhalten implementiert. FE II berechnet eine zur Vorgabe aus FE I passende, kollisionsfreie Trajektorie
Trennung der Ebenen	IV.c / IV.d	IV / V

²⁸⁵ Dieses Beispiel wird auch aufgeführt in Sukthankar (1997): Situation Awareness for Tactical Driving, S. 75

²⁸⁶ s. dazu auch Lotz & Winner (2014): Maneuver Delegation and Planning for Automated Vehicles...

Tabelle 5-6: Umsetzung der Referenzsituationen in der Führungsebene

		V1	V2
1	Fahrstreifen befahren mit vorausfahrendem Fzg.	Manöver "Fahrstreifen folgen" würde von FE I an FE II übermittelt werden. FE II entscheidet über Fahrzeugfolgeverhalten und berechnet eine kollisionsfreie Trajektorie. FE II berücksichtigt dabei externe Einflussfaktoren (aus Weltmodell: Zul. Höchstgeschw.; Aus Fahrersicht: Komfortkriterien).	Analog zu V1
2	Fahrstreifenwechsel zulassen	Die Notwendigkeit sowie die Ausführung einer Anpassung der Fahrzeuglängsführung, um einen Fahrstreifenwechsel eines anderen Fahrzeugs zuzulassen, würde rein in FE II erkannt bzw. ausgeführt werden.	Die Notwendigkeit einer Anpassung der Fahrzeuglängsführung, um einen Fahrstreifenwechsel eines anderen Fahrzeugs zuzulassen, würde in FE I erkannt; Die Anpassung der Längsführung würde von FE I an FE II beauftragt werden mit Hilfe der Vorgabe geeigneter Ausführungsparameter und deren Toleranzbereich.
3	Annähern an Knoten / Fußgängerüberweg	Weitergabe der räumlichen Endbedingung des Manövers "Fahrstreifen folgen" von FE I an FE II ist erforderlich, um auf Basis dieser Information die Längsführung in FE II an die Situation (einschl. Vorfahrtsregelung) anpassen zu können, z. B. um die Geschwindigkeit im Vorfeld zu reduzieren.	Parametrierung der Längsführung erfolgt analog zu Referenzsituation 2
4	Koop. / takt. Fahrstreifenwechsel in Lücke	<p>Da die Entscheidung über Ausführung eines koop./takt. Fahrstreifenwechsels nur in Abhängigkeit von einem bestimmten Manöververlauf (zulässiger Wertebereich von fahrdynamischen Systemzustandsgrößen) getroffen werden kann und letzterer durch die Schnittstellenwahl nicht kommuniziert werden kann, so bestehen die Möglichkeiten:</p> <p>a) Koop./takt. Manöver sind nicht möglich bzw. müssen statt vom System (FE I) vom Fahrer initiiert werden</p> <p>b) FE I prüft zur Laufzeit den Nutzen eines koop./takt. Manövers und beauftragt es ggf. an FE II. Dort ist ein ähnlicher Algorithmus implementiert, der eine dazu passende kollisionsfreie Trajektorie berechnet</p> <p>c) In FE II wird ständig der Nutzen eines koop./takt. Fahrstreifenwechsels innerhalb eines dafür vorgesehenen, eigenen Verhaltens geprüft und ggf. an FE I vorgeschlagen. FE I prüft, ob der Fahrstreifenwechsel ein Element der Manöver zur Zielerreichung darstellt und entscheidet auf dieser Basis über Annahme / Ablehnung des Vorschlags.</p> <p>Das Vorbereiten des Fahrstreifenwechsels, d.h. die Anpassung der Längsführung an eine Lücke im Nachbar-Fahrstreifen, würde im Verantwortungsbereich von FE II liegen.</p>	Entscheidung über Nutzen eines koop./takt. Manövers wird in FE I getroffen. Da eine solche Entscheidung mit einem dafür zulässigen Wertebereich der Manöververlaufs verknüpft ist, werden diese in Form eines Toleranzbereichs an FE II gegeben. FE II plant darauf aufbauend eine kollisionsfreie Trajektorie und optimiert diese im Rahmen des gegebenen Entscheidungsspielraums. Sollte aus Sicht von FE II keine sichere Ausführung des Fahrstreifenwechsels möglich sein, so wird die Ausführung verzögert, bis ggf. eine Neuplanung von FE I erfolgt. Das Vorbereiten des Fahrstreifenwechsels würde demnach im Verantwortungsbereich von FE I liegen.

Tabelle 5-6: Umsetzung der Referenzsituationen in der Führungsebene (Fortsetzung)

		V1	V2
5	Koop./takt. Fahrstreifenwechsel bei Annäherung an Knotenpunkt	Fahrstreifenwechsel wird in b) durch Entscheidung in FE I verhindert, in c) durch Rückkopplung FE II -> FE I (s. Referenzsituation 4)	Da die Entscheidung zur Ausführung eines Manövers in FE I getroffen wird und dort die zulässigen Manöver zur Zielerreichung bekannt sind, wird ein koop./takt. Fahrstreifenwechsel vor einem Knotenpunkt verhindert.
6	Annähern an Kreuzung: Erreichen eines Abbiege-Fahrstreifens bzw. Einordnen in Fahrstreifen	Der vorbereitende Fahrstreifenwechsel muss durch Weitergabe der räumlichen Endbedingung von FE I sichergestellt werden. Der sich daraus ergebende Entscheidungsspielraum über Beginn und Verlauf des Fahrstreifenwechsels nutzt FE II, um eine passende und kollisionsfreie Trajektorie zu berechnen.	Die Weitergabe des Ausführungsspielraums von FE I an FE II enthält die Information, bis wann ein vorbereitender Fahrstreifenwechsel ausgeführt sein muss.
7	Vorfahrt Gewähren beim Abbiegen / Kreuzen / in Kreisverkehr Einfahren	Das Abbiegemanöver wird auf Basis räumlicher Nähe und unter Berücksichtigung von Lichtsignalanlagen zum Einfahrtspunkt der Kreuzung von FE I an FE II beauftragt. Der eigentliche Beginn des Abbiegens, also das Einhalten des Vorfahrt-Gebots, wird, wie auch der Verlauf des Manövers, in FE II entschieden.	FE I entscheidet über durchzuführendes Fahrmanöver sowie dessen Ausführungsparameter, einschließlich des Beginns (d.h. Einhaltung des Vorfahrts-Gebots). FE II plant Trajektorie im Rahmen des vorgegebenen Toleranzbereichs.

In Variante 1 würde die FE I in den Referenzsituationen 1 und 2 lediglich das Verhalten „Fahrstreifen folgen“ an die FE II beauftragen. Das reaktive Abstandhalten zu einem Vorderfahrzeug sowie das kooperative Verhalten des Einfädelnlassens müsste deshalb in FE II abgebildet werden. Eine Änderung bzw. Anpassung der Längsführung führt in beiden Fällen nicht zu einer Verhaltensanpassung auf Manöverebene.

In Referenzsituation 3 nähert sich das automatisierte Fahrzeug einem Entscheidungspunkt zur Fortsetzung der Längsführung an, wie beispielsweise einem Fußgängerübergang oder einer Kreuzung. In diesen Situationen kann es - abhängig von Kreuzungsgeometrie, Vorfahrtsregelung und eigener Manöverintention - notwendig sein, die Fahrzeuglängsführung in Vorbereitung an eine mögliche Verhaltensänderung, wie bspw. das Halten, Abbiegen oder Kreuzen, anpassen zu müssen. Geyer²⁸⁷ verbildlicht dieses Problem durch ein Tor („Gate“), dem sich das Fahrzeug nähert und das solange verschlossen ist, bis ein regelkonformes und gefahrloses Durchfahren der danach folgenden Situation sichergestellt ist. Ein solches Verhalten kann weder als rein reaktiv bezeichnet werden, da es auch ohne jegliche Beteiligung anderer Verkehrsobjekte notwendig ist (z. B. bei Annäherung an eine Lichtsignalanlagen-geregelte Kreuzung), noch als rein

²⁸⁷ Geyer (2013): Entwicklung und Evaluierung eines kooperativen Interaktionskonzepts...

deliberativ (die zur Zielerreichung notwendige Manöverfolge ändert sich dadurch nicht). Fiele die Aufgabe des Annäherns der FE I zu, so müsste diese die Position der Entscheidungspunkte an die FE II übertragen, sodass die FE II darauf aufbauend eine Trajektorie planen kann, die den Anforderungen des Entscheidungspunktes gerecht werden (z. B. das rechtzeitige Verzögern und Anhalten). Alternativ müsste die FE II selbst auf Basis des aktuellen Manövers / Fahrstreifens die Entscheidungspunkte ableiten. Da jedoch die Existenz von Entscheidungspunkten an Kreuzungen auch vom auszuführenden Manöver zur Zielerreichung abhängt (bspw. das Vorfahrtachten beim Linksabbiegen, das beim Kreuzen nicht notwendig wäre), wird die zweitgenannte Option aus Konsistenzgründen verworfen.

Bezugnehmend auf Referenzsituation 4 ist die Aufteilung gemäß Variante 1 bisher nicht ausreichend definiert, da die Entscheidung zur Verhaltensänderung (Entscheidung über Manöver-Änderung von „Fahrstreifen folgen“ zu „Fahrstreifenwechsel“) stark abhängig von den Ausführungsparametern des Manövers selbst ist. Im Hinblick auf Tabelle 5-3 bedeutet das eine starke Kopplung der Ebenen IV.c / d., also genau entgegen der Auftrennung, die Variante 1 versucht. Hinsichtlich der Umsetzung von Variante 1 ergeben sich mehrere denkbare Untervarianten. In Untervariante 1.a sind kooperative / taktische Manöver nicht umsetzbar oder müssen statt vom System vom Fahrer initiiert werden, was jedoch das direkte Ausscheiden der Variante zur Folge hat (Nicht-Erreichung der funktionalen Anforderungen). In Untervariante 1.b besitzt entgegen der ursprünglichen Aufteilung die FE I die Fähigkeit, nicht nur deliberative Manöver zu planen, sondern auch den Gesamtnutzen eines kooperativen / taktischen Manövers zu prüfen, wobei dieses im positiven Fall an die FE II beauftragt wird. Ohne eine Aufweitung der Schnittstelle zwischen FE I und FE II inkl. einer Übertragung definierter Ausführungsparameter (siehe Ebenen IV.d.1 - IV.d.4 in Tabelle 5-3), besteht jedoch kein effektiver Kontrollmechanismus zwischen FE I und FE II. Dies steht im Gegensatz zu einer Variante 1.c, in der stattdessen in FE II der Nutzen eines kooperativen / taktischen Fahrstreifenwechsels, bspw. im Rahmen eines eigens dafür implementierten Verhaltens, ermittelt wird und im positiven Falle an die FE I vorgeschlagen wird. FE I kann dann entsprechend des dort abgelegten Entscheidungsspielraums - der zur Zielerreichung möglichen Manöver - überprüfen, ob das vorgeschlagene Manöver der Zielerreichung widerspricht und dann akzeptieren oder ablehnen. Durch diese Rückkopplung verbindet Variante 1.c die deliberativen Systemkomponente mit der reaktiven, kooperativen / taktischen. Dieser Mechanismus löst ebenfalls Referenzsituation 5, in dem das vorgeschlagene Manöver von FE I nicht akzeptiert wird.

In Referenzsituation 6 muss das System das Erreichen eines zur Zielerreichung notwendigen Abbiege-Fahrstreifens sicherstellen. Eine ähnliche Situation stellt auch das Einfädeln in einen durchgehenden Fahrstreifen von einem endenden Fahrstreifen dar (s. Tabelle 2-1: Fahrstreifenwechsel unter Zwang). Aus diesem Grund muss FE I zumindest die räumliche Endbedingung für einen vorbereitenden Fahrstreifenwechsel an FE II

weiterreichen, was analog zur Diskussion zur Referenzsituation 3 eine Änderung / Erweiterung der vorher definierten Schnittstelle von IV.c / IV.d auf IV.d.1 zur Folge hätte.

Das Vorfahrtachten wird in Referenzsituation 7 betrachtet, in der das automatisierte System an einer Kreuzung einem oder mehreren anderen Fahrzeugen die Vorfahrt gewähren muss. In Variante 1 wird das Vorfahrtachten als ein reaktives Systemelement betrachtet (z. B. das Abbiegen in eine geeignete Lücke) und ist somit FE II zuzuschreiben (Berechnung einer kollisionsfreien Fahrzeugtrajektorie).

Die zweite im Rahmen dieses Abschnittes betrachtete Architekturvariante spricht der FE I im Vergleich zu Variante 2 mehr Entscheidungskompetenz zu, sodass sich die Trennung von FE I und FE II durch die Ebenen IV (Entscheidung über Ausführung des Manövers) und V (Ausführung des Manövers) gemäß Tabelle 5-3 ergibt. Diese Variante wird in Tabelle 5-5 (rechte Seite) aufgeführt. Sie ist durch eine kombinierte deliberative, kooperative / taktische Manöverplanung in FE I und eine reaktive Trajektorienplanung in FE II charakterisiert. Als Schnittstelle zwischen FE I und FE II ist demnach das jeweilige aktuell auszuführende Fahrmanöver zu übermitteln sowie die ggf. situationsbedingt notwendigen Ausführungsparameter, d.h. Werte für Systemzustandsvariablen, die während der Ausführung oder am Ende des Manövers eingehalten werden müssen (vgl. Tabelle 5-3, Ebene V.a). Dabei können ggf. vorhandene Entscheidungsspielräume der höheren Systemebenen in FE I an die FE II „vererbt“ werden (z. B. zulässige Geschwindigkeitsbereiche bei Manöver-Ende, raumzeitliche Spielräume der Ausführung). Entscheidungsspielräume können dabei auch verallgemeinert als Kosten angesehen werden, die im Zusammenhang mit der Erreichung des Zielzustandes stehen und von der FE II nicht überschritten werden dürfen.

In Referenzsituation 1 wird dabei das (reaktive) Abstands- und Folgeverhalten zu einem vorausfahrenden Fahrzeug analog zu Variante 1 von der FE II bewerkstelligt. Da es sich gemäß Kap. 5.1.2 beim Manöver „Fahrstreifen folgen“ um ein sog. implizites Manöver handelt, das nur durch den Beginn eines expliziten Manövers (bspw. Fahrstreifenwechsel) oder das Fahrmissions-Ende beendet wird, ist eine Weitergabe von Ausführungsparametern an FE II nur eingeschränkt möglich und notwendig. Denkbar in diesem Zusammenhang wäre jedoch der zulässige Geschwindigkeitsbereich oder, ähnlich eines ACC-Systems, eine Soll-Zeitlücke zum vorausfahrenden Fahrzeug.

Das kooperative Einfädelnlassen anderer Fahrzeuge in Referenzsituation 2 würde gemäß der Variantenbeschreibung der FE I zukommen. Die Erkenntnis einer notwendigen Anpassung der Längsführung, bspw. durch die Existenz eines Fahrzeuges auf einem benachbarten Einfädelstreifen, erfordert bereits ein fortgeschrittenes Szenenverständnis sowie ein darauf angepasstes Situationsprädiktionsmodell und unterscheidet sich demnach deutlich von klassisch reaktiven Systemelementen. Die Ausprägung der Anpassung der Längsführung müsste jedoch in der Schnittstelle von FE I an FE II berücksich-

tigt werden und kann sich von den Parametern aus Referenzsituation 1 unterscheiden (z. B. Zeitlücke in Bezug auf das Fahrzeug auf dem Nachbarfahrstreifen). Das Annähern an einen Entscheidungspunkt in der Längsführung in Referenzsituation 3 wird dabei in gleicher Weise gelöst, da auch hierzu zunächst eine reine Anpassung der Längsführung notwendig ist.

Ein eigener Fahrstreifenwechsel in eine Lücke gemäß Referenzsituation 4 wird in Variante 2 gelöst, in dem die FE I die Nützlichkeit bzw. Notwendigkeit eines Fahrstreifenwechsels ermittelt und im Bedarfsfall unter Weitergabe der dazu gültigen Ausführungsparameter (bzw. max. zulässige Kosten) an die FE II übermittelt. Im Gegensatz zu den vorherigen Situationen stellt der Fahrstreifenwechsel dabei ein explizites Manöver mit definierbaren Anfangs- und Endbedingungen dar. Als Schnittstelle zur FE II können und müssen somit die zulässigen Systemzustandsvariablen zu Beginn, während und zum Ende des Fahrstreifenwechsels übertragen werden. Dies ist notwendig, da die Entscheidung zur Ausführung des Manövers nur für eine definierte Ausführung gültig ist.

Das Verhindern eines Fahrstreifenwechsels vor einem Knotenpunkt in Referenzsituation 5 wird intern in FE I bewerkstelligt, da dort sowohl die zur Zielerreichung zulässigen Manöver ermittelt als auch die Entscheidung zur Manöverdurchführung von Fahrstreifenwechsel getroffen werden. Das rechtzeitige Einordnen bzw. Erreichen eines Abbiegefahrstreifens in Referenzsituation 6 wird analog zu vorherigen Situationen durch Beauftragung des Manövers von FE I an FE II erreicht, inkl. der Weitergabe der raumzeitlichen Endbedingungen des Manövers als Spielraum für die Trajektorienplanung in FE II.

Das Vorfahrtachten aus Referenzsituation 7 wird insbesondere in FE I abgebildet, da hier die Manöver, einschließlich Abbiegen / Kreuzen, initiiert werden. Dennoch besitzt die FE II wie auch in den bereits zuvor diskutierten Referenzsituationen die Aufgabe, im Rahmen des vorgegebenen Spielraums eine kollisionsfreie Trajektorie zu planen. Somit wird ersichtlich, dass die Hierarchieaufteilung gemäß Variante 2 sowohl in FE I als auch in FE II eine Kollisionsfreiheit des aktuellen und zukünftigen Verhaltens mit anderen Verkehrsteilnehmern zum Ziel hat. Explizite Fahrmanöver werden demnach nur dann tatsächlich ausgeführt, wenn sowohl FE I als auch FE II eine Kollisionsfreiheit sicherstellen können. Dies bedeutet aber auch, dass sich beide Ebenen widersprechen können, z. B. dass trotz Beauftragung von FE I die Ausführung eines Manövers in FE II verzögert, verhindert oder abgebrochen wird, d.h. allgemein die von FE I akzeptierten Kosten zur Durchführung eines expliziten Manövers von FE II nicht eingehalten werden können. Dementsprechend muss in FE I stets ein Alternativplan vorhanden sein, der in diesem Fall an FE II beauftragt wird²⁸⁸. Umgekehrt muss FE II in der Lage sein, einen

²⁸⁸ Anmerkung: Eine solche Alternative kann auch bereits explizit über die Schnittstelle von FE I an FE II übermittelt werden. Um die Schnittstellen möglichst "schlank" zu halten, wird diese Option nicht weiter vertieft (Kriterium Nr. 7 in Tabelle 2-2).

Abbruch eines Manövers nach Vorgabe von FE I umzusetzen. Letzteres ist als eine funktionale Anforderung zu bezeichnen, die sich jedoch auch in der Schnittstellendefinition zwischen den beiden Ebenen widerspiegelt. Dies impliziert, dass eine Rückführung des aktuellen Ausführungsstatus von FE II an FE I notwendig ist, um die notwendigen Planungsschritte der beiden Ebenen im Rahmen der zulässigen Entscheidungsspielräume konsistent und synchron zu halten.

Im nächsten Abschnitt werden die Kriterien, die zur Bewertung der zwei Varianten herangezogen werden, näher erläutert.

Bewertungskriterien

Wie bereits im vorherigen Abschnitt erwähnt, ist die Konsistenz der Beschreibung der Architekturvarianten über möglichst alle zur Laufzeit auftretenden Situationen von besonderer Bedeutung für die Architekturbewertung und entspricht der Forderung nach hoher semantischer Kohäsion gemäß Tabelle 2-2. Hierfür liefert die im vorherigen Abschnitt durchgeführte, Variantenbildung und -beschreibung auf Basis der Referenzsituationen eine Bewertungsgrundlage.

Darüber hinaus besteht auch bei der Bewertung der Architekturvarianten zur Auftrennung der Führungsebene analog zu Abschn. 5.2.2 die Forderung nach Einhaltung der Grundsätze hybrider Schichtenarchitekturen, insbesondere nach Abbildung unterschiedlicher Abstraktionsgrade innerhalb der Ebenen (unterschiedlich überblickter raumzeitlicher Horizont, unterschiedlich lange Entscheidungsperioden).

Weitere Bewertungskriterien, die sich heranziehen lassen, sind die Prüfung auf Möglichkeit der Dekomposition mit der Frage, ob zwischen Entscheidung und Ausführung von Systemfunktionen unterschieden wird (Kriterium Nr. 3 in Tabelle 2-2) sowie die Prüfung auf Refaktorisierung mit der Frage, ob Zuständigkeiten nicht redundant vorliegen oder widersprüchlich sind (Kriterium Nr. 12).

Variantenbewertung

Variante 1: Deliberative Manöverplanung, reaktive Trajektorienplanung

Variante 1 versucht die aus der Robotik bekannte, klassische Aufteilung von deliberativen und reaktiven Systemelementen in Führungsebene I durch ein „Aktionsauswahlmodul“ (bzw. in diesem Fall Verhaltens- bzw. Manöverauswahlmodul) darzustellen, das Handlungsfolgen zur Zielerreichung in eine Sequenz von Steuerungsparametern zerlegt, die sich auf die reaktiven Verhaltensbausteine abbilden lassen (vgl. Kap. 3.3). Die propagierte Aufteilung einer rein deliberativ motivierten Manöverauswahl ohne Berücksichtigung der Manöverausrührung stößt jedoch bereits ohne Beteiligung anderer Verkehrsobjekte an ihre Grenzen, da sich insbesondere bei der Annäherung an

Knotenpunkte verhaltensrelevante Einschränkungen ergeben, die an die untergeordnete Führungsebene II kommuniziert werden müssen (siehe Referenzsituationen 3 und 6). Dieser Umstand wird in Situationen verschärft, die eine komplexe Interaktion mit anderen Verkehrsobjekten beinhalten, wie bspw. beim kooperativen / taktischen Fahrstreifenwechsel in dichtem Verkehr (s. Referenzsituation 4). Die dafür notwendige Kopplung zwischen der Entscheidung zur Durchführung eines Manövers und der Entscheidung über die Ausführung kann Variante 1 nicht abbilden. Aus diesem Grund kann festgestellt werden, dass sich die Entscheidungskompetenzen und die damit verbundenen notwendigen Schnittstelleninformationen zwischen FE I an FE II nicht konsistent über alle betrachteten Situationen hinweg beschreiben lassen, was als negativ zu bewerten ist. Zusätzlich entsteht in Variante 1c (Vorschlag möglicher kooperativer / taktischer Manöver an FE I) eine „aufwärts“ gerichtete Schnittstelle entgegen der eigentlichen Richtung der Entscheidungs-Hierarchie ohne eigene Kontrollmöglichkeit durch FE I.

Wird in Variante 1 der Informationsbedarf von FE I und FE II analysiert, so kann festgestellt werden, dass FE I im Gegensatz zu FE II keine Information über die Existenz und Zustände anderer Verkehrsobjekte und nur eine abstrahierte Repräsentation des statischen Umfeldes benötigt, d.h. eine abstrahierte Repräsentation (z. B. Graph und Eigenposition) der fahrstreifengenauen Szenerie. Auch überblickt FE I einen größeren raumzeitlichen Horizont (bis zum nächsten Knotenpunkt) als FE II (max. bis zum nächsten Knotenpunkt), was als positiv zu bewerten ist. Semantisches Kontextwissen, wie bspw. Verkehrsregel-relevante Merkmale wie Verkehrszeichen und daran abgeleitete Verkehrsregeln, werden jedoch in beiden Ebenen benötigt (s. Referenzsituation 3 und 7).

Variante 1 unterscheidet zwischen Entscheidung und Ausführung (Prüfung auf Möglichkeit zur Dekomposition). Die Prüfung auf mögliche Refaktorisierung lässt sich jedoch nicht eindeutig beantworten: Zwar lässt sich die Zuständigkeit von FE I so definieren, dass sie die Entscheidung über das auszuführende Manöver trifft, jedoch ist sie effektiv für die Entscheidung zur Ausführung kooperativer / taktischer Manöver auf den Vorschlag einer anderen Instanz (FE II) angewiesen, was faktisch zusätzliche Abhängigkeiten nach außen schafft und in negativer Weise auch die Kriterien 4 (hohe Kohäsion), 5 (lose Kopplung) und 8 (Separation-of-Concerns) gemäß Tabelle 2-2 betrifft.

Variante 2: Deliberative, kooperative / taktische Manöverplanung, reaktive Trajektorienplanung

Variante 2 führt in FE I die Manöverplanung nicht nur auf Basis deliberativer Intention durch, sondern ist auch in der Lage, taktische bzw. durch kooperatives Verhalten motivierte Manöverplanungen selbstständig auszuführen. Dies ist deswegen als vorteilhaft zu bewerten, da die Aufgabe und Zuständigkeit von FE I (und damit auch FE II) über

alle diskutierten Referenzszenarien konsistent bleibt und die Architekturvariante damit im Vergleich zu Variante 1 eine höhere semantische Kohäsion besitzt.

Dieser Vorteil wird jedoch dadurch „erkauft“, dass das Problem der Koordination deliberativer und reaktiver Systemanteile nicht durch zwei eindeutig voneinander getrennte Module abgebildet wird, sondern teilweise in das Innere eines Moduls verlagert und somit „versteckt“ wird. Dies trifft für Variante 2 zu, da die Reaktivität des Gesamtsystems nicht nur rein in FE II abgebildet wird, sondern durch die Zuständigkeit, kooperative / taktische Fahrmanöver zu initiieren oder Vorfahrt zu achten, implizit auch in FE I. Aus diesem Grund muss die FE I nachfolgend nochmals näher betrachtet werden.

Unabhängig von der Art der Implementierung der Ebenen lassen sich dabei, basierend auf Tabelle 5-3, folgende Feststellungen ableiten:

- Die deliberative Systemkomponente äußert sich dadurch, dass räumliche Einschränkungen hinsichtlich der aktuellen und zukünftig erlaubten Aufenthaltsbereiche des eigenen Fahrzeugs existieren, die im Rahmen der gewählten Hierarchieaufteilung in Form von (erlaubten) Zielfahrstreifen von der Navigations- an die Führungsebene übergeben werden (vgl. Abschn. 5.2.2). Anhand der Position des Eigenfahrzeugs im bekannten, fahrstreifengenauen Umfeldmodell lassen sich auf dieser Basis die für das Fahrzeug zur Zielerreichung zulässigen Fahrmanöver bestimmen.
- Die in FE I zu berücksichtigende reaktive Systemkomponente als Teil des kooperativen / taktischen Verhaltens muss in der Lage sein, die aktuelle Ausführung eines Fahrmanövers beeinflussen oder ggf. eine Änderung des Manövers bewirken zu können.

Letztlich lassen sich beide Anliegen auf zwei Arten kombinieren:

- Einerseits kann ein in FE I implementiertes Manöverplanungs-Submodul (z. B. für kooperative / taktische Fahrstreifenwechsel) der dort ebenfalls implementierten deliberativen Systemkomponente das aus seiner Sicht beste Manöver vorschlagen, die den Vorschlag annehmen oder ablehnen kann. Diese Ausprägung wäre ähnlich zu Variante 1c, wobei jedoch die beiden interagierenden Submodule innerhalb einer Hierarchieebene lägen.
- Die deliberative Systemkomponente kann einem Manöverplanungs-Submodul in FE I den zulässigen Entscheidungsspielraum vorab entweder in Form der erlaubten Aufenthaltsbereiche (Zielfahrstreifen) oder der erlaubten Fahrmanöver mitteilen.

Insbesondere die zweite Kombinationsmöglichkeit besitzt den Vorteil eines rückführungsfreien Informationsflusses sowie der Verhinderung unnötiger (weil unplausibler) Planungsaktivitäten. Ebenfalls bietet sie die Möglichkeit, indirekt durch Vorgabe nur eines verbleibenden erlaubten Manövers ein Manöver durchzusetzen, dass entweder zur

Zielerreichung notwendig ist oder - wie im teilautomatisierten Modus analog zur PRORETA 3 Funktion gefordert wird - vom Fahrer als Vorgabe an das automatisierte System delegiert wurde. Die zweite Kombinationsmöglichkeit bietet somit ein über alle betrachtete Referenzsituationen konsistentes Erklärungsmodell des Zusammenwirkens deliberativer und reaktiver Systemelemente, was als vorteilhaft zu bewerten ist.

Im Gegensatz zu Variante 1 benötigen sowohl FE I und FE II Informationen über die aktuellen und zukünftigen Zustände dynamischer Objekte, jedoch nicht zwangsläufig im gleichen Detaillierungsgrad. So ist eine Konfiguration von FE I denkbar, die nur auf eine abstrahierte Umfeldmodell-Repräsentation auf Basis eines Graphen der umgebenden Fahrstreifen zugreift, einschließlich einer räumlichen Zuordnung der Verkehrsobjekte zu diesen. Darüber hinaus benötigt FE I zusätzlichen Zugriff auf Verkehrsregelrelevante Merkmale wie bspw. Verkehrszeichen, um einen semantischen Kontext der Situation aufbauen und die Verkehrsobjekte darin und zueinander zuordnen zu können. Letztere Information muss für FE II nicht zur Verfügung stehen, da bereits alle Verhaltensanforderungen, die sich aus dem situativen Gesamtkontext ergeben, in Form der beauftragten Manöver einschließlich der situativ relevanten Entscheidungsspielräume an FE II kommuniziert wurden. Dies kann als besonders positiv hervorgehoben werden, da so die in FE II implementierte Verhaltensausführung von der Notwendigkeit des Besitzes von situationsspezifischem Kontextwissen und eines komplexen Situationsverständnisses entlastet wird und so besser den Grundsätzen hybriden Architekturen mit eher „klassisch“ reaktiven Verhalten auf der unteren Hierarchieebene vereinbar ist. FE II kann somit auf dieser Basis und unter Kenntnis der genauen Bewegungszustände der anderen Verkehrsobjekte eine komfortable, aber in jedem Fall kollisionsfreie Fahrzeugtrajektorie planen. Anhand des Informationsbedarfs kann festgestellt werden, dass FE I im Vergleich zu FE II abstrahierte Informationen verarbeitet und, analog zu Variante 1, einen größeren raumzeitlichen Horizont überblickt, was als positiv zu bewerten ist.

Auch Variante 2 unterscheidet prinzipiell zwischen Verhaltensentscheidung in FE I und -ausführung in FE II, wobei die Prüfung auf Möglichkeit der Dekomposition zu verneinen und somit ebenfalls als positiv zu bewerten ist. Dennoch ist die Trennung durch eine teilweise „Vorwegnahme“ des Ausführungsergebnisses durch FE I im Vergleich zu Variante 1 weniger stark ausgeprägt. Analog ergibt die Prüfung auf Refaktorisierung, dass die Zuständigkeit der Sicherstellung der Kollisionsfreiheit sowohl in FE I als auch in FE II verankert ist, was zunächst als negativ zu bewerten ist.

Auf den zweiten Blick ergeben sich damit jedoch zusätzliche Möglichkeiten hinsichtlich der Absicherung automatisierter Fahrzeugführungssysteme. Hillenbrand²⁸⁹ bezeichnet eine solche Anordnung, die auf der Übereinstimmung von Ergebnissen zur Verhinderung unerwünschter Ausführungen von Systemfunktionen beruht, als eine strukturelle

²⁸⁹ Hillenbrand (2012): Funktionale Sicherheit nach ISO 26262..., S. 182 ff

Redundanz. Diese kann im Kontext der ISO 26262 auch zur Dekomposition von funktionalen Sicherheitsanforderungen auf mehrere Module verwendet werden, in diesem Fall auf FE I und FE II. Dies ist jedoch nur dann möglich, wenn die beiden Module ausreichend unabhängig voneinander sind. Dies ist dann hinreichend der Fall, wenn beide Module auch von unterschiedlichen Umfeldmodellen, respektive Umfeldsensoren, gespeist werden.

Variantenauswahl und Diskussion

Im Rahmen der vorliegenden Arbeit wird Variante 2 zur Auftrennung der Führungsebene präferiert. Dies liegt insbesondere daran, dass in Variante 2 die Aufgabe der Manöverplanung in einer Hierarchie-Ebene isoliert wird und nicht - im Gegensatz zu Variante 1 - versucht wird, die Aufgabe der Verhaltens- bzw. Manöverplanung hinsichtlich der zugrundeliegenden Intention aufzutrennen (deliberativ einerseits, kooperativ / taktisch andererseits). Letzterer Umstand resultiert in einer schlechten Erklärbarkeit der Zuständigkeiten, Schnittstellen und Abhängigkeiten der beiden Ebenen FE I und FE II über verschiedene Verkehrssituationen hinweg.

Im Gegensatz dazu findet in Variante 2 in FE I eine situationsübergreifende Planung der vom System durchzuführenden Verhalten im Sinne von Fahrmanövern statt und bildet so die Anforderung ab, die Entscheidung zu einer Verhaltensänderung mit einer spezifischen Erwartung an die Ausführung zu koppeln. Die Notwendigkeit dieses Umstandes wird besonders deutlich durch Referenzszenario 4 (kooperativer / taktischer Fahrstreifenwechsel), wobei die Entscheidung zu einem Fahrstreifenwechsel nur dann getroffen werden kann, wenn gleichzeitig eine Annahme zur Beschleunigung bzw. Verzögerung des eigenen Fahrzeugs gegenüber der für die Verhaltensdurchführung zuständiger Instanzen durchgesetzt werden kann. Effektiv muss die Führungsebene I damit eine Kopplung von Fahrzeuglängs- und Querverführung berücksichtigen, was prinzipiell der Aufgabe der Trajektorienplanung in FE I sehr ähnlich ist. Deshalb besteht für Variante 2 umgekehrt die „Gefahr“, sämtliche komplexen Vorgänge, nämlich insbesondere die Kopplung deliberativer und reaktiver Systemelemente einerseits sowie die Entscheidung über und Ausführung von Verhalten andererseits in einer monolithischen Struktur zusammenzufassen.

In der Literatur existieren vielfach Ansätze, die das Problem der Situationsanalyse, Manöver- und Trajektorienplanung algorithmisch geschlossen lösen, jedoch auf einzelne Funktionen oder Anwendungsdomänen beschränkt sind. So beschreiben bspw. Bender & Stiller²⁹⁰ einen kombinierten Manöver- und Trajektorienplanungsansatz für Über-

²⁹⁰ Bender & Stiller (2015): Trajektorienplanung: Manöveridentifikation anhand der Topologie...

holszenarien auf Landstraßen oder Schwarting & Pascheka²⁹¹ einen integrierten Ansatz zur kooperativen Manöver- und Trajektorienplanung auf Autobahnen. Um das Problem einer monolithischen Struktur zu vermeiden, muss im Rahmen der Architekturentwicklung Wert darauf gelegt werden, die einzelnen „nicht-trennbaren“ Einheiten in der Gesamtwirkkette zu identifizieren und herauszulösen, in diesem Fall auch innerhalb von FE I selbst.

Die im vorherigen Abschnitt empfohlene nochmalige Unterteilung der FE I in ein deliberatives Teilsystem, das dem kooperativen / taktischen Teilsystem die zur Zielerreichung zulässigen Planungsspielräume in Form von (Ziel-) Fahrstreifen bzw. zulässigen Fahrmanövern einschließlich der räumlichen Endbedingungen weiterreicht, schafft eine Möglichkeit zur FE I-internen Auftrennung der deliberativen Systemkomponente.

Dabei ist es denkbar, dass die deliberative Komponente von einer Instanz begleitet wird, die diskrete und regelbasierte Verhaltenseinschränkungen, z. B. das Anhalten an einer Lichtsignalanlage mit roter Phase oder das Einhalten eines Überholverbots, an den „flexibleren“ Teil der Manöverplanung weitergegeben kann. Dies wäre auch im Sinne der Testbarkeit und der Forderung nach möglichst deterministischem Systemverhalten (vgl. Kap. 2.3.2) als positiv zu bewerten. Diese Auslegung ergänzt die von Arkin²⁹² propagierte Entscheidungsstrategie der *Selektion* (die deliberative Komponente bestimmt, welche Verhaltenselemente und deren Parameter zur Laufzeit ausgeführt werden) um den zulässigen *Verhaltensspielraum* und schafft somit eine konsistente Schnittstelle sowohl FE I-intern als auch zwischen FE I und FE II. Hinsichtlich des Kontrollflusses ergeben sich somit zwischen FE I und FE II zwei wichtige Aspekte: Zum einen müssen Verhaltensspielräume möglichst über alle Situationen hinweg in konsistenter Art und Weise an FE II übermittelt werden und zum anderen muss eine Rückführung des Ausführungsstatus von FE II an FE I erfolgen, um die erwartungskonforme Ausführung der Verhalten überprüfen zu können.

Beispiele für verhaltensrelevante Spielräume im Kontext automatisierten Fahrens sind vielfach in der Literatur erwähnt. So kann das bereits erwähnte „Gate“-Konzept nach Geyer²⁹³, das in ähnlicher Form auch Broggi et al.²⁹⁴ vorstellen, als eine Möglichkeit zur Übermittlung einer solchen Information angesehen werden. Ebenfalls sind Konzepte, die sog. „Fahrschläuche“ bzw. Fahrkorridore als Optimierungsgrundlage für unterla-

²⁹¹ Schwarting & Pascheka (2014): Recursive Conflict Resolution for Cooperative Motion Planning...

²⁹² Arkin (2000): Behavior-Based Robotics, S. 214

²⁹³ Geyer (2013): Entwicklung und Evaluierung eines kooperativen Interaktionskonzepts...

²⁹⁴ „A gate integrates information about speed, position and orientation that the vehicle must have in the future trajectory without directly plan an entire trajectory“. Broggi (2013): Moving from analog to digital driving, S. 1115

gerte Trajektorienplanungsmodule vorgeben, denkbar (siehe z. B. Ziegler et al.²⁹⁵). Ein weiteres, sehr anschauliches Beispiel für das Zusammenspiel zwischen Verhaltensentscheidung und -ausführung mit enger Kopplung liefert Pellkofer^{296a}: Dabei übergibt die Verhaltensentscheidung zeitfunktionale Sollverläufe mit Toleranzbereichen von Systemzustandsgrößen an die für die Verhaltensausführung verantwortliche Instanz^{296b}. Die Toleranzbereiche sind so definiert, dass sie bei erwartungskonformem Ablauf der Aktionsausführung nicht verlassen werden. Je schmaler dabei der Toleranzbereich ist, desto strenger formuliert die Verhaltensentscheidung die Erwartungen an den Verlauf der Aktion. Hinsichtlich der Prüfung auf eine erwartungskonforme Ausführung schreibt Pellkofer^{296c}: „*Da es bei Aktionen zu Fehlern und Verzögerungen kommen kann, muss für eine robuste Ausführung der Status und Fortschritt der Aktionen in geeigneter Weise rückgekoppelt werden. Dazu ist es notwendig, dass die Experten [Anm.: Mit Experte ist ein Modul zur Verhaltensausführung gemeint] den Fortschritt der bei ihnen ablaufenden Aktionen bemessen und anzeigen. Dabei benutzt jeder Experte sein detailliertes Expertenwissen dazu, sinnvolle Maße für den Fortschritt seiner Aktionen zu generieren*“.

Das Fortschrittsmaß leitet sich dabei vom Ist-Verlauf der o.g. System-Zustandsgrößen ab. Verlässt der Ist-Verlauf den vorher definierten Toleranzbereich, so kann die Verhaltensentscheidung angemessen reagieren (z. B. Abbruch des Verhaltens bzw. des Manövers). Die Rückführung des Ausführungsstatus untergeordneter Hierarchieebenen an höhere entspricht dabei den Grundsätzen hybrider Basisarchitekturen (siehe dazu auch Hertzberg et al.²⁹⁷) und stellt ein notwendiges Mittel dar, um mit Unsicherheiten der aktuellen Umfeldinformationen und zukünftigen Situationsentwicklung umgehen zu können.

Insgesamt lässt sich somit ein konsistentes Gesamtbild hinsichtlich Zuständigkeiten und Schnittstellen der Systemhierarchie auf Basis von Variante 2 gewinnen. Eine detailliertere Beschreibung der Zusammenhänge zwischen FE I und FE II - oder im Allgemeinen zwischen Verhaltensentscheidung und Ausführung im Kontext automatisierten Fahrens - ist des Weiteren nicht möglich, ohne auf implementierungs- oder Anwendungsdomänenspezifische Details eingehen zu müssen, und steht somit nicht im Fokus der funktionalen Systemarchitektur. Hinweise zur Anwendung der in dieser Arbeit aufgeführten Gestaltungsgrundsätze einschließlich der Nutzung von Fahrmanövern als Verhaltens-elemente können jedoch, neben den bereits in diesem Abschnitt erwähnten Quellen, u.a.

²⁹⁵ Ziegler et al. (2014): Kartengestütztes automatisiertes Fahren auf der Bertha-Benz-Route...

²⁹⁶ Pellkofer (2003): Verhaltensentscheidung für autonome Fahrzeuge mit Blickrichtungssteuerung. a: -; b: S. 72; c: S. 67

²⁹⁷ Hertzberg et al. (2012): Mobile Roboter, S. 322 + 325

den Arbeiten von Hakuli et al.²⁹⁸, Dickmanns²⁹⁹, Maurer³⁰⁰ und Siedersberger³⁰¹ entnommen werden.

Im Rahmen der vorliegenden Arbeit ergeben sich somit insgesamt vier Architektur-Hierarchieebenen, wobei die Führungsebene aufgrund ihres im Vergleich zur übergeordneten Navigations- bzw. untergeordneten Stabilisierungsebene deutlich höheren Funktionsumfangs nochmals unterteilt wird. Eine graphische Übersicht inkl. einer Zusammenfassung der Schnittstellenbeschreibung der sich damit ergebenden vollständig definierten Kontrollhierarchie wird in Kap. 6.1 gegeben.

Auf Basis der festgelegten Hierarchie können nun die Informations- und Kommunikationsbedarfe der Planungsmodule hinsichtlich Weltmodell und Mensch-Maschine-Schnittstelle abgeleitet werden. Bevor dies erfolgt, muss jedoch noch ein weiterer wichtiger Systembestandteil betrachtet werden - die Fähigkeit zu entscheiden, in welchem Betriebsmodus sich das automatisierte Fahrzeugführungssystem befindet.

5.3 Modus-Entscheidung und -Überwachung

Die bisherigen Ausführungen im Rahmen dieser Arbeit basieren auf der Annahme, dass sich das System im aktiven Zustand befindet und alle Voraussetzungen, die eine automatisierte Fahrt ermöglichen, erfüllt sind. Dadurch ist ersichtlich, dass ein automatisiertes Fahrzeugführungssystem über mindestens zwei diskret benennbare Betriebsmodi verfügt, die in simpler Weise als „aktiviert“ und „deaktiviert“ bezeichnet werden können. Darüber hinaus ist jedoch eine größere Anzahl weiterer Betriebsmodi denkbar, die stark von der jeweilig betrachteten Granularität sowie der Zielsetzung des Gesamtsystems abhängig sind. So sind im Ausführungsbeispiel von PRORETA 3 drei Betriebsmodi vorgesehen: Das manuelle (nicht assistierte) Fahren, der Modus *Sicherheitskorridor* (eingreifende Notfallfunktion) sowie der Modus *kooperative Automation* (teilautomatisiert). Denkbar wäre, dazu zukünftig noch einen Modus „vollautomatisiertes Fahren“ hinzuzufügen (vgl. Abschn. 5.2.2). Dadurch wird ersichtlich, dass der Betriebsmodus auch an den Automationsgrad des Gesamtsystems gekoppelt ist. Die Bezeichnungen und Ausprägungen von Betriebsmodi unterliegen dabei keiner allgemeingültigen Regelung. Im Kontext des automatisierten Fahrens könnten jedoch zukünftig regulatorische Faktoren, die auch Rückschlüsse auf rechtliche Fragen zulassen (z. B. die Einteilung

²⁹⁸ Hakuli et al. (2010): Development and Validation of Manoeuvre-Based Driver Assistance Functions...

²⁹⁹ Dickmanns (2007): Dynamic Vision for Perception and Control of Motion

³⁰⁰ Maurer (2000): Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen

³⁰¹ Siedersberger (2003): Komponenten zur automatischen Fahrzeugführung...

von Automationsgraden gemäß Gasser et al.³⁰² oder der SAE Norm J3016³⁰³) dazu führen, dass definierte Funktionsumfänge des Gesamtsystems in diskreten Betriebsmodi zusammengefasst werden müssen.

Im Hinblick auf den Stand der Technik adressiert Maurer^{304a} den Bedarf, den *Grad der Automatisierung* und damit den Betriebsmodus zur Laufzeit flexibel ermitteln und einstellen zu können (vgl. Kap. 3.4.1). Welchen Betriebsmodus (respektive Automationsgrad) das Gesamtsystem einnimmt, hängt gemäß Maurer^{304b} insbesondere von drei Faktoren ab:

- Der aktuellen Leistungsfähigkeit des Gesamtsystems,
- dem Fahrerwunsch sowie
- der Domäne, in der sich das System aktuell befindet (z. B. Autobahn).

Ein zentrales Element in der Arbeit von Maurer stellt das *Fähigkeitenkonzept* dar, das zusammen mit Pellkofer³⁰⁵ und Siedersberger³⁰⁶ erarbeitet wurde. Dabei wird in Fähigkeiten zur Wahrnehmung, Entscheidung, Planung und Aktion unterschieden (wobei letztere auch als Verhaltensweisen bezeichnet werden)^{304c}. Je nach Ausführungsqualität insbesondere der Wahrnehmungs- und Aktionsfähigkeiten kann der Grad der Automation erhöht oder muss dieser verringert werden (*funktionale Degradation*). Maurer entwickelt in seiner Arbeit Gütemaße, mit denen diese Ausführungsqualität zur Laufzeit quantifiziert werden kann und sieht in seiner Architektur ein Modul zur Überwachung der Leistungsfähigkeit vor, das diese Gütekriterien ermittelt und einer zentralen Entscheidungsinstanz (*CD: Central Decision*), die auch den Grad der Automatisierung ermittelt, bereitstellt^{304d}. Verschlechtern sich bspw. die Fähigkeiten zur Wahrnehmung, so kann es notwendig sein, dass der Fahrer während einer automatisierten Fahrt die Kontrolle zurück übernehmen muss, was nicht nur für ergonomische, sondern auch rechtlichen Fragestellungen von großer Bedeutung ist.

Werden die Erkenntnisse aus der Literatur als funktionale Zuständigkeiten an ein solches Modul zusammengetragen, so ergeben sich diese wie folgt: Ein Modus-Entscheidungs- und Überwachungsmodul, im nachfolgenden kurz *Modusplaner* genannt, vermittelt zwischen gewünschtem Betriebsmodus des Fahrers / Passagiers einerseits und aktuell situations- und systembedingt zulässigem Betriebsmodus andererseits.

³⁰² Gasser et al. (2012): Rechtsfolgen zunehmender Fahrzeugautomatisierung

³⁰³ SAE International (2014): Standard J3016 - Taxonomy and Definitions...

³⁰⁴ Maurer (2000): Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen. a: -; b: S. 4; c: S. 61; d: S. 96

³⁰⁵ Pellkofer (2003): Verhaltensentscheidung für autonome Fahrzeuge mit Blickrichtungssteuerung

³⁰⁶ Siedersberger (2003): Komponenten zur automatischen Fahrzeugführung..., S. 63 ff

Das Modul stellt somit ein Gleichgewicht zwischen externen (Betriebs-) Anforderungen und interner Leistungsfähigkeit her. Situationsbedingte Einschränkungen des Betriebsmodus können sich dabei

- aus der aktuellen Verkehrsdomäne ergeben,
- aufgrund extern verursachter Einschränkungen der Leistungsfähigkeit (z. B. durch widrige Witterungsbedingungen)
- oder durch plötzliches oder vorhersehbares Auftreten bestimmter Situationsattribute bzw. deren Konstellation, die das System nicht unterstützt (z. B. plötzlich: Wegfall von Straßenmarkierungen; vorhersehbar: Nähern einer Mautstelle o.ä.).

Systembedingte Einschränkungen des Betriebsmodus können durch

- (technischen) Defekt, Fehler und (Teil-) Ausfällen von Hard- und Softwarekomponenten,
- funktionale Unzulänglichkeiten, einschließlich nicht eigenständig lösbarer Konflikte innerhalb der Planungsmodule der Entscheidungshierarchie oder durch das
- Fehlen von notwendigen Betriebsmitteln des Fahrzeugs (z. B. Kraftstoff / elektr. Energie) verursacht werden.

Weitere „systembedingte“ Einschränkungen können sich durch Indikatoren für den geschätzten Fahrerzustand³⁰⁷ ergeben, z. B. hinsichtlich Aufmerksamkeit, Müdigkeit oder für einen möglichen Fall von Zweckentfremdung bzw. Missbrauchs des Systems (z. B. Überwachung von Lenkmoment, Gurtschloss).

Der Modusplaner initiiert bei jedem Übergang von Betriebsmodi die dazu notwendigen Modulfunktionalitäten sowie Verhaltensweisen³⁰⁸. Eine wichtige Anforderung in diesem Zusammenhang ist, dass bei der Verwaltung der Betriebsmodi und deren Übergänge zu jederzeit ein Zustandsübergang in einen risikominimalen Systemzustand erfolgen können muss. In welcher Art und Weise dieser erfolgen muss, hängt vom ursprünglichen Betriebsmodus und insbesondere vom ursprünglichen Automationsgrad ab. So wäre dies bspw. bei einem heutig erhältlichen Fahrerassistenzsystem mindestens eine Systemdeaktivierung, die im Fehlerfall als *fail-silent* ausgelegt ist, bei einem zukünftig hochautomatisiertem Fahrzeug jedoch mindestens ein Rückführung in den sog. risikominimalen Zustand³⁰⁹, der in jedem Fall als *fail-operational* auszulegen ist³¹⁰.

Als weitere Voraussetzung für ein solches Modul müssen Schnittstellen von und zum Fahrer (MMS) vorliegen, die in der Lage sind, eine Vermittlung zwischen gewünschtem

³⁰⁷ s. dazu auch Langer et al. (2015): Fahrerzustandserkennung

³⁰⁸ s. dazu auch Maurer (2000): Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen, S. 97

³⁰⁹ vgl. Gasser et al. (2015): Rahmenbedingungen für die Fahrerassistenzsystementwicklung, S. 31

³¹⁰ Bahram et al (2015): Please Take Over!...

und aktuellem bzw. möglichem Betriebsmodus zu leisten. Dazu gehören insbesondere die folgenden Informationen:

- An MMS: Anzeige der aktuell zulässigen Betriebsmodi
- An MMS: Anzeige des aktiven Betriebsmodus
- Von MMS: Eingang des aktuell gewünschten Betriebsmodus / bzw. Übergang
- An MMS: Ggf. Zusatzinformationen hinsichtlich notwendiger Übergänge von Betriebsmodi und deren Begründungen

Darüber hinaus müssen dem Modusplaner eingangsseitig alle Informationen der überwachten Systemzustände vorliegen (s. obige Liste der situations- und systembedingten Einschränkungen). Die Informationen können dabei bereits vorausgewertet sein, wie bspw. die Indikatoren für die Leistungsfähigkeiten wichtiger System-Fähigkeiten (insbes. hinsichtlich Weltmodellqualität). Aus Modularitätsgründen wird jedoch empfohlen, die Eingangsinformationen möglichst funktionsunabhängig zu gestalten, um so im Modusplaner an zentraler Stelle den Gesamtsystem-Funktionsumfang spezifizieren zu können³¹¹.

Die Verhaltensänderung des Systems unter Berücksichtigung der Leistungsfähigkeit der einzelnen Wahrnehmungs- und Aktionsfähigkeiten lässt sich - abgesehen von den genannten diskreten Betriebsmodi und deren Übergänge - detailliert modellieren. So muss bspw. eine Verschlechterung der Leistungsfähigkeit der Umfelderkennung nicht zwangsläufig in einer rein „reaktiven“ Änderung bzw. Degradation des übergeordneten Betriebsmodus resultieren, sondern kann auch durch eine „pro-aktive“ Anpassung des Fahrverhaltens bzw. der Fahrstrategie entschärft werden (analog zum menschlichen Verhalten, bspw. bei verringerter Sichtweite langsamer zu fahren oder auf die Durchführung bestimmter Manöver, wie z. B. Fahrstreifenwechsel, zu verzichten). Eine solche Erweiterung der Modul-Zuständigkeit führt jedoch ebenfalls eine Schnittstellenerweiterung an die Planungsmodule mit sich, die in einer Beeinflussung der Entscheidung und Ausführung von Verhaltensweisen resultiert (z. B. das Absenken der max. erlaubten Fahrgeschwindigkeit bei verringerter Sichtweite).

Zusammenfassend ist festzuhalten, dass im vorliegenden Abschnitt die Notwendigkeit eines Modus-Entscheidungs- und Überwachungsmoduls hergeleitet und die funktionalen Zuständigkeiten eines solchen Moduls benannt wurden. Es verbleibt jedoch die Frage, wie ein solches Modusplanungs-Modul konkret in die in diesem Kapitel hergeleitete Entscheidungshierarchie eingebettet werden kann.

³¹¹ Die Leistungsfähigkeit einer Systemkomponente / eines Moduls sollte möglichst objektiv und funktionsunabhängig zur Verfügung gestellt werden. Bsp.: Erkennungsreichweite der kamerabasierten Fahrstreifenerkennung in Meter statt der Aussage, ob die Reichweite für einen bestimmten Betriebsmodus ausreicht oder nicht.

Maurer^{312a} bspw. setzt sein als „zentrale Entscheidungsinstanz“ benanntes Modul zwischen das für die Navigationsaufgabe zuständige „Missionsplanungsmodul“ und die „Planungsmodule für die Fortbewegung“ (ähnlich zu den in dieser Arbeit bezeichneten Führungsebene-Modulen)³¹³. Diese Entscheidung äußert sich dadurch, dass Maurer seine Entscheidungshierarchie als hierarchischen Zustandsautomaten implementiert, in der die zentrale Entscheidungsinstanz als oberste Ebene des Automaten implementiert wurde^{312b}.

Einer solche Konfiguration wird im Rahmen dieser Arbeit nicht entsprochen, vor allem da die Entscheidungshierarchie in den vorherigen Abschnitten des vorliegenden Kapitels bereits vollständig hergeleitet und begründet wurde. Eine Einbettung des Modusplanungsmoduls in die definierte Hierarchie oder sogar innerhalb eines der vier erarbeiteten Planungsmodule (z. B. in die Führungsebene) ist im Hinblick auf die deutlich unterschiedlichen funktionalen Zuständigkeiten (Fahrzeugführungsaufgabe versus Modus-Entscheidung und -Überwachung) als nicht empfehlenswert zu bezeichnen (weil es den in Tabelle 2-2 genannten Kriterien 5: Lose Kopplung, 8: Separation-of-Concerns; und 13: Semantische Kohäsion widerspricht). Aus den genannten Gründen wird das Modul im Rahmen der vorliegenden Arbeit entsprechend als eigenständiges Modul betrachtet, wobei der Modusplaner jedoch direkten Einfluss auf die Planungsmodule ausüben kann.

5.4 Weltmodell

Der Begriff *Weltmodell* (engl. *World Model*) beschreibt im Kontext der Robotik eine Struktur, die ein Modell des aktuellen und zukünftigen internen und externen Umfelds des Systems generiert, repräsentiert und so Planungsmodulen zugänglich macht. Damit beinhaltet es eine Repräsentation des Fahrzeugumfeldes, der eigenen Systemzustände sowie ggf. auch die des Fahrers^{314 312c}, was in der Literatur oft unter dem Begriff *Wissensrepräsentation* (oder engl. *knowledge representation*) zusammengefasst wird³¹⁵.

³¹² Maurer (2000): Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen. a: S. 46 + 96ff; b: S. 52 ff; c: S. 58 ff

³¹³ Die Schnittstelle zwischen zentraler Entscheidungsinstanz und Missionsplanung wurde jedoch nicht detailliert beschrieben, was eine Bewertung der Übertragbarkeit erschwert.

³¹⁴ s. z. B. Reichardt (1996): Kontinuierliche Verhaltenssteuerung eines autonomen Fahrzeugs..., S. 30ff

³¹⁵ Eine Einführung zum Thema Wissensrepräsentation im Kontext der Robotik gibt Arkin (2000): Behavior-Based Robotics, S. 177ff

Häufig wird dabei in drei verschiedene Arten der Repräsentationen unterschieden, nämlich *geometrische*, *semantische* und *topologische* Repräsentationen^{316 317 318}.

Eine ausführliche Übersicht über geometrische Umfeldrepräsentationen, wie bspw. (2D) Grid Maps oder (3D) Gitterkarten, ist u.a. in Schreier³¹⁹ sowie Burgard & Hebert³²⁰ gegeben.

Semantisches Wissen repräsentiert den Situationskontext, in dem sich das System befindet, und wird dadurch gewonnen, dass (geometrische) Objekte (oder auch *Entitäten*) des statischen und dynamischen Umfeldes miteinander in Relation gesetzt werden. So kann z. B. durch Zuordnung von anderen Fahrzeugen oder auch Verkehrszeichen zu erkannten Fahrstreifen eine Aussage abgeleitet werden (*Inferenz*), welche Situationsaspekte für die eigene Verhaltensentscheidung relevant sind (z. B. „langsamer fahrendes Fahrzeug voraus“) und ebenso das zukünftige Verhalten anderer Fahrzeug geschätzt werden. Zusätzlich können dadurch für das eigene System nicht relevante Informationen gezielt vernachlässigt werden, was zu einer Entlastung der Planungsmodule führt (Sukthankar³²¹ bezeichnet dies als *selektive Perzeption*). In der Literatur existieren eine Vielzahl von sog. *Ontologien*, die eine formale Spezifikation von Begriffen und Regeln zur *expliziten*³²² (und insbesondere semantischen) Wissensrepräsentation darstellen³²³. Anschauliche Beispiele für Ontologien zur Wissensrepräsentation in einer Verkehrssituation geben u.a. Stiller & Lulcheva³²⁴ mit Ansätzen des probabilistisch-logischen Lernens, Pellkofer^{325a} mittels linguistischer Variablen auf Basis der Fuzzy-Theorie und Kohlhaas et al.³²⁶ mit einer semantischen Zustandsraumdarstellung.

Topologische Repräsentationen abstrahieren Zusammenhänge des Umfeldes durch Bildung von Knoten und Kanten sowie deren Attribute. So wird das Straßennetz häufig

³¹⁶ s. z. B. Reichardt (1996): Kontinuierliche Verhaltenssteuerung eines autonomen Fahrzeugs..., S. 11

³¹⁷ s. z. B. Kohlhaas et al (2014): Semantic State Space for High-Level Maneuver Planning...

³¹⁸ s. z. B. Matthaei (2015): Wahrnehmungsgestützte Lokalisierung in fahrstreifengenauen Karten..., S. 38

³¹⁹ Schreier (2016): Bayesian Environment Representation, Prediction, and Criticality..., S. 8ff

³²⁰ Burgard & Hebert (2008): World Modeling

³²¹ Sukthankar (1997): Situation Awareness for Tactical Driving, S. 26

³²² Im Gegensatz zu *impliziten* bzw. *taziten* (verborgenen) Repräsentation, vgl. Arkin (2000): Behavior-Based Robotics, S. 181

³²³ Definition von Ontologie, Taxonomie: Siehe Stiller & Lulcheva (2008): Probabilistische Inferenz für Kognitive Automobile, S. 3

³²⁴ Stiller & Lulcheva (2008): Probabilistische Inferenz für Kognitive Automobile

³²⁵ Pellkofer (2003): Verhaltensentscheidung für autonome Fahrzeuge.... a: S. 50 ff; b: S. 47

³²⁶ Kohlhaas et al. (2014): Semantic State Space for High-Level Maneuver Planning...

als topologische Karte repräsentiert, in dem die Kanten Straßenzüge darstellen und die Knoten Knotenpunkte wie z. B. Kreuzungen. Zusätzliche Attribute können geometrische und auch semantische Informationen an Knoten und Kanten anhängen, wie z. B. Position, Weglänge und Verkehrsregeln³²⁷ (vgl. auch Abschn. 5.2.1).

Im Hinblick auf die vorliegende Arbeit ist nun von besonderem Interesse, welche Repräsentationsarten für die bereits ermittelten Planungsmodule notwendig sind und im Weltmodell explizit repräsentiert würden müssen. Hierzu schreibt Pellkofer^{325b}:

„Da es eine Vielzahl von Situationsaspekten gibt und diese auf eine noch größere Anzahl von Objektkombinationen angewandt werden können, ist eine grundsätzliche Berechnung aller Situationsaspekte ineffizient und bei detaillierter werdender Situationsbeschreibung in Echtzeit nicht durchführbar. Deshalb kann und soll die Berechnung nur solcher Situationsaspekte erfolgen, die von der Verhaltensentscheidung benötigt und weiterverarbeitet werden. Ob eine Situationsbeschreibung ausreichend detailliert ist, hängt davon ab, welche Fähigkeiten dem System zur Verfügung stehen und welche davon überhaupt in Frage kommen“.

Ob Wissen explizit im Umfeldmodell repräsentiert werden muss oder im Gegensatz dazu nur implizit vorliegt und ggf. in den Planungsmodulen berechnet bzw. implizit weiterverarbeitet wird, hängt somit stark vom betrachteten Funktionsumfang und damit auch der Verkehrsdomäne ab, in der das System eingesetzt wird. Eine allgemeingültige Aussage über notwendige Informationen und somit Schnittstellen zu den Planungsmodulen ist somit nicht möglich³²⁸. Darüber hinaus sind notwendige Schnittstelleninformationen zwischen Weltmodell und Planungsmodulen stark abhängig vom jeweilig eingesetzten Algorithmus und somit letztendlich implementierungsabhängig. Aus diesen Gründen werden im Folgenden nur die wichtigsten Repräsentationsarten aufgeführt, die für das automatisierte Fahren in strukturierten Umgebungen notwendig sind und aus denen sich möglichst vollständig alle weiteren semantischen Zusammenhänge generieren lassen³²⁹.

Eine sehr ausführliche Quelle zu diesem Problem stellt Matthaei^{330a} dar, der im Kontext des Projektes Stadtpilot (s. Kap. 3.4.3) ebenfalls eine Architektur für automatisiertes Fahren erstellt, jedoch mit Schwerpunktsetzung auf das Weltmodell und Nutzung von

³²⁷ vgl. Matthaei (2015): Wahrnehmungsgestützte Lokalisierung in fahrstreifengenauen Karten..., S. 7

³²⁸ vgl. Burgard & Hebert (2008): World Modeling, S. 853

³²⁹ „We emphasize the appropriate use of knowledge, not merely using knowledge for knowledge’s sake“, aus Arkin Arkin (2000): Behavior-Based Robotics, S. 178

³³⁰ Matthaei (2015): Wahrnehmungsgestützte Lokalisierung in fahrstreifengenauen Karten.... a: -; b: S. 48; c: S. 44; d: S. 52 ff

Kartendaten. Da die Ergebnisse weitestgehend übertragbar sind, sei für eine tiefergehende Betrachtung deswegen auf die genannte Quelle verwiesen.

Notwendige Eingangsgrößen für die im Rahmen dieser Arbeit hergeleitete Navigationsebene ist vor allem die topologische Karte des Straßennetzes mit zusätzlichen geometrischen und semantischen Informationsattributen^{330b}, um die Aufgabe der Routenplanung³³¹ einschließlich der Navigationsanweisung durchführen zu können. Matthaei bezeichnet die topologische Karte als ein externes Datum, das das eigene System nicht zur Laufzeit mittels eigener Umfeldsensorik ermittelt, sondern ihm zur Laufzeit mittels Funkverbindung³³² oder a-priori über einen Datenträger zur Verfügung gestellt wird. Weitere optionale externe Informationen zur Verbesserung der Funktion können z. B. den Verkehrsfluss oder verkehrsrelevante Ereignisse (z. B. Stau, Unfall) sein. Da externe Daten zwecks Wiederverwendbarkeit immer in einem globalen Koordinatensystem referenziert sind^{330c}, muss ebenfalls die absolute, globale Eigenposition bekannt sein und der Navigationsebene nach erfolgtem Abgleich als kartenrelative Pose übermittelt werden.

Deutlicher umfangreicher ist die notwendige Schnittstelle zwischen Weltmodell und Führungsebene, insbesondere da hier im Gegensatz zur Navigationsebene auch Informationen über das dynamische Umfeld des Fahrzeugs notwendig sind. Eine wichtige Repräsentationsform ist dabei die topologische und geometrische Beschreibung der Strukturelemente der statischen Verkehrsszenerie, die Matthaei als *fahrstreifengenaue Weltmodellierung* beschreibt. Als Beispiel für eine fahrstreifengenaue Weltmodellierung beschreiben Czerwionka et al.³³³ eine Karte, die auf sog. Route Network Definition Files (RNDF)³³⁴ aufbaut, die Rahmen der DARPA Urban Challenge verwendet wurden (vgl. Kap. 3.4.2). Abbildung 5-3 zeigt ein Beispiel einer fahrstreifengenaue Karte eines kommerziellen Kartenproviders im Vergleich zu einer straßengenauen Navigationskarte. Wie Matthaei zeigt^{330d}, ist es jedoch auch möglich, eine solche Karte mit verringerter Reichweite zur Laufzeit aus Informationen der bordeigenen Umfeldsensorik zu schätzen und mit a-priori Karten zu fusionieren. Wichtig in diesem Zusammenhang sind auch das Generieren sowie die Weitergabe von semantischen Informationen, die die Verkehrsregeln betreffen, wie z. B. Geschwindigkeitsbegrenzungen oder Vorfahrtsregelungen, die als Attribut an die topologischen Elemente der Karte abgehängt werden können.

³³¹ siehe dazu Kleine-Besten et al. (2015): Navigation und Verkehrstelematik

³³² z. B. über ein sog. *Backend*, siehe Klanner & Ruhhammer (2015)

³³³ Czerwionka et al. (2012): Optimized Route Network Graph as Map Reference...

³³⁴ [http://archive.darpa.mil/grandchallenge/docs/RNDF MDF Formats 031407.pdf](http://archive.darpa.mil/grandchallenge/docs/RNDF_MDF_Formats_031407.pdf), letzter Zugriff am 03.10.16



Abbildung 5-3: Unterschied zwischen straßen- und fahstreifengenauen (HD-) Karten³³⁵

Weitere Repräsentationsformen, die zur automatisierten Fahrzeugführung notwendig sind, sind weniger stark abstrahierte, metrische Umfeldmodelle zur Beschreibung von belegten und freien Bereichen des Fahrzeugumfelds³³⁶. Diese können bspw. dann notwendig sein, wenn Umfeldmodelle höheren Abstraktionsgrades, wie die o.g. fahstreifengenaue Karten, zur Laufzeit nicht in der notwendigen Validität zur Verfügung stehen (z. B. fehlende / nicht erkannte Markierungen) oder die Strukturelemente „Fahstreifen“ nicht als maßgeblich zur Manöver- bzw. Trajektorienplanung erachtet werden (z. B. beim Notbremsen bzw. Notausweichen). Wie bereits erwähnt, werden auch Informationen über das dynamische Umfeld benötigt, insbesondere Position und deren zeitliche Ableitung von anderen Verkehrsobjekten sowie ergänzenden Attributen wie z. B. Typ, räumliche Ausdehnung etc.. Gemäß Matthaei^{337a} werden die aufgeführten Informationen bereits fahrzeugrelativ an die Führungsebenen-Module weitergegeben, weshalb im Weltmodell bereits eine fahstreifengenaue Lokalisierung stattfinden muss.

Notwendige Eingangsgrößen für die Stabilisierungsebene (d.h. die Trajektorienregelung) sind vor allem die quasi-kontinuierliche Informationen des Eigensystemzustands einschließlich einer hochgenauen lokalen Pose^{337b}.

Eine weitere Zuständigkeit des Weltmodells ist, wie bereits im vorherigen Abschnitt beschrieben, die Zurverfügungstellung der Kennwerte der eigenen Leistungsfähigkeit. Darüber hinaus wird nahezu unabhängig vom Funktionsumfang oder Automationsgrad automatisierter Fahrzeuge eine Prädiktion des Bewegungsverhaltens der anderen Verkehrsobjekte benötigt, wobei in der Literatur häufig in Trajektorien- sowie Manöverprädiktion unterschieden wird. An dieser Stelle sei dazu auf Schreier³³⁸ verwiesen, der

³³⁵ <https://company.here.com/en/automotive/intelligent-car/here-hd-map/>, letzter Zugriff am 30.03.15

³³⁶ vgl. Schreier (2016): Bayesian Environment Representation, Prediction, and Criticality...

³³⁷ Matthaei (2015): Wahrnehmungsgestützte Lokalisierung in fahstreifengenauen Karten.... a: S. 41; b: S. 51

³³⁸ Schreier (2016): Bayesian Environment Representation, Prediction, and Criticality..., Kap. 6

entsprechende Ansätze für Trajektorien- und Manöverprädiktion im Kontext von PRORETA 3 entwickelt und implementiert hat.

Umgekehrt ist auch ein Informationsfluss von den Planungsmodulen an das Weltmodell denkbar. So diskutiert Pellkofer^{339a} ausführlich den Themenbereich des *aktiven* bzw. *aufgabenbezogenen Sehens* als gegenseitige Abhängigkeit von visueller Aufgabe und aktivem Handeln (unabhängig vom verwendeten Sensorkonzept wird auch von Aufmerksamkeitssteuerung gesprochen). Darauf aufbauend beschreibt Pellkofer die Implementierung des sog. EMS-Vision Systems³⁴⁰ nach Vorbild von Dickmanns³⁴¹, das eine bewegliche Kameraplattform beinhaltet, die je nach situativer Anforderung unterschiedliche Bereiche im erreichbaren Blickfeld fokussieren kann. Das Aktivieren gezielter Wahrnehmungsfähigkeiten auf Basis der Erfordernisse des Missionsziels kann in diesem Zusammenhang auch als *vision for action* bezeichnet werden³⁴². Umgekehrt ist ein Verhalten denkbar, das auch als *action for vision* bezeichnet werden kann, also eine gezielte Beeinflussung der Bewegungsplanung des Fahrzeugs, um bspw. Defizite des Erfassungsbereichs von Sensoren auszugleichen oder in Bereiche einsehen zu können, die durch andere Objekte verdeckt werden (Bsp.: Analog zum menschlichen Verhalten des „Hineintastens“ in unübersichtliche Kreuzungen). Die beiden genannten Aspekte werden in der vorliegenden Arbeit jedoch nicht weiter vertieft, da

- komplexe Wechselwirkungen zwischen planenden Modulen und Weltmodell bzw. Sensorik geschaffen werden, die zu einem überwiegenden Teil ökonomisch motiviert (bspw. um Sensorinfrastruktur einzusparen)^{339b} und somit nicht Schwerpunkte der Arbeit sind,
- Wechselwirkungen zur Hardwarearchitektur bestehen (Nachteilig in Bezug auf Kriterium Nr. 9 in Tabelle 2-2),
- bei einer Implementierung einer Verhaltensweise *action for vision* eine zyklische Abhängigkeit zwischen Weltmodellierung und Planung geschaffen wird, die es gemäß den nichtfunktionalen Anforderungen der vorliegenden Arbeit zu vermeiden gilt (Kriterium Nr. 1 in Tabelle 2-2).

³³⁹ Pellkofer (2003): Verhaltensentscheidung für autonome Fahrzeuge.... a: -; b: S. 8

³⁴⁰ EMS steht dabei für Expected-based Multi-focal Saccadic Vision

³⁴¹ vgl. Dickmanns et al. (1994): The Seeing Passenger Car „VaMoRs-P”

³⁴² Pellkofer (2003): Verhaltensentscheidung für autonome Fahrzeuge..., S. 18, nach Barattoff et al. (1999): Ortsvariantes aktives Sehen...

6 Synthese

Im folgenden Abschnitt werden die gewonnenen Erkenntnisse zusammengefasst und anschließend bzgl. der Erfüllung der funktionalen Anforderungen aus Kap. 2 gegenübergestellt (Abschn. 6.2). In Abschn. 6.3 erfolgt darüber hinaus eine Beschreibung einer beispielhaften Umsetzung der Architekturelemente im Rahmen des Projekts PRORETA 3.

6.1 Bausteinsicht

Abbildung 6-1 zeigt die oberste Ebene der Bausteinsicht der Referenzarchitektur auf Basis der Ausführungen der vorherigen Kapitel. Sie bildet die wichtigsten Systembausteine und Schnittstellen ab. Es sei darauf hingewiesen, dass die Schnittstellenbezeichnungen in Abbildung 6-1 nicht alle tatsächlich übertragenen Informationen benennen. Eine Detaillierung hierzu kann Anhang B entnommen werden.

Die Architektur besteht aus vier hierarchisch angeordneten Architektur-Ebenen bzw. Schichten, wobei neben der Navigations- und Stabilisierungsebene die in Kap. 5.2.4 beschriebene Zweiteilung der Führungsebene explizit dargestellt wird. Die zu jeder Ebene partitionierten Systemfunktionen gemäß Kap. 5.2 werden stellvertretend in jeweils einem Modul repräsentiert und als *Missionsplaner*, *Verhaltensplaner*, *Trajektorienplaner* und *Trajektorienregler* benannt. Die vier Module bilden untereinander eine *organisatorische Hierarchie*, die Maurer³⁴³ als ein Modell beschreibt, in dem obere Entscheidungsinstanzen unterlagerten Instanzen zwar koordinieren, ihnen jedoch eigene Entscheidungsspielräume zugestehen (vgl. Kap. 5.2.4). Diese Entscheidungsspielräume spielen eine zentrale Rolle in der entwickelten Referenzarchitektur und sind bereits in Tabelle 5-3 benannt.

Das in der Navigationsebene angeordnete und als *Missionsplaner* bezeichnete Entscheidungs- und Planungs-Modul dient dazu, auf Basis einer topologischen Repräsentation des Straßennetzes und der vom Fahrer³⁴⁴ vorgegebenen Fahrmission eine möglichst optimale Fahrtroute zu planen. Konkrete Entscheidungsspielräume ergeben sich somit in der Auswahl der zu befahrenden Straßensegmente aus einem Graphen, der das Stra-

³⁴³ Maurer (2000): Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen, S. 33 nach Mesarovic et al. (1970): Theory of Hierarchical, Multi-Level Systems

³⁴⁴ Anm.: Bei fahrerlosen Systemen könnte der Begriff „Operator“ angemessener sein

Bennetz repräsentiert. Im Gegensatz zur klassischen Definition nach Donges wird die Funktion der Ermittlung der Referenz auf die zulässigen Zielfahrstreifen vor und nach Entscheidungspunkten (als eine genauere Beschreibung der Navigationsanweisung) der Navigationsebene und somit dem Missionsplaner zugeordnet (vgl. Kap. 5.2.2).

Die Aufgabe des in der Führungsebene I allokierten *Verhaltensplaners* ist es, eine Sequenz von Verhaltensweisen auszuwählen, die die Erreichung der definierten Zielfahrstreifen unter Berücksichtigung der anderen Verkehrsobjekte sowie der geltenden Verkehrsregeln ermöglicht. Somit findet im Verhaltensplaner die komplexe Aufgabe der Kombination von deliberativen und reaktiven Systemelementen statt. Die Verhaltensweisen wurden im Rahmen der vorliegenden Arbeit als Fahrmanöver gemäß Tabelle 2-1 definiert. Wie in Kap. 5.2.4 beschrieben, ist die Entscheidung, ein bestimmtes Manöver durchzuführen und somit als Element der Verhaltenssequenz auszuwählen, speziell in Verkehrssituationen, die eine Interaktion mit anderen Verkehrsteilnehmern bedürfen, an eine bestimmte Erwartung an die Ausführung dieses Manövers gekoppelt. Somit besitzt die Entscheidung zur Ausführung einen definierten Gültigkeitsbereich hinsichtlich des zukünftigen eigenen Verhaltens. Dieser Gültigkeitsbereich muss dementsprechend an die unterlagerten Ebenen, in diesem Fall an den in Führungsebene II angeordneten Trajektorienplaner in geeigneter Form kommuniziert werden (vgl. Kap. 5.2.4), wobei dieser wiederum dem Trajektorienplaner als Verhaltensspielraum gilt.

Neben der Erwartung an das eigene Verhalten ist eine Manöverentscheidung ebenfalls vom prognostizierten Verhalten anderer Verkehrsobjekte sowie der zeitlichen Konsistenz der Verkehrsszenerie abhängig. Dementsprechend muss als Eingangsschnittstelle für den Verhaltensplaner seitens des Weltmodells zum einen die aktuelle Verkehrsszene als Momentaufnahme der Zustände von statischem und dynamischen Umfeld sowie darüber hinaus eine Prädiktion des Verhaltens des dynamischen Umfelds übermittelt werden, um daraus eine Prognose über das gesamte Verkehrsszenario³⁴⁵ einschließlich des eigenen Verhaltens ermitteln zu können.

Eine denkbare Rückkopplung an den Missionsplaner wäre das Übermitteln aktuell unzulässiger Zielfahrstreifen (z. B. aufgrund einer anhaltenden Nicht-Erreichbarkeit durch Baustellen / Blockaden), sodass eine Routen-Neuplanung erfolgen kann. Als eigener Entscheidungsspielraum besteht für den Verhaltensplaner die grundsätzlich freie Planung von Fahrmanövern, sofern sie die Erreichbarkeit der Zielfahrstreifen nicht gefährden (die sich z. B. in einer freien Fahrstreifenwahl auf einer Autobahn fern von der geplanten Abfahrt äußert), wobei das Beachten der Verkehrsregeln vorausgesetzt wird.

³⁴⁵ an dieser Stelle sei nochmals auf die Begriffsdefinition verwiesen in Geyer et al. (2014): Concept and Development of a Unified Ontology...

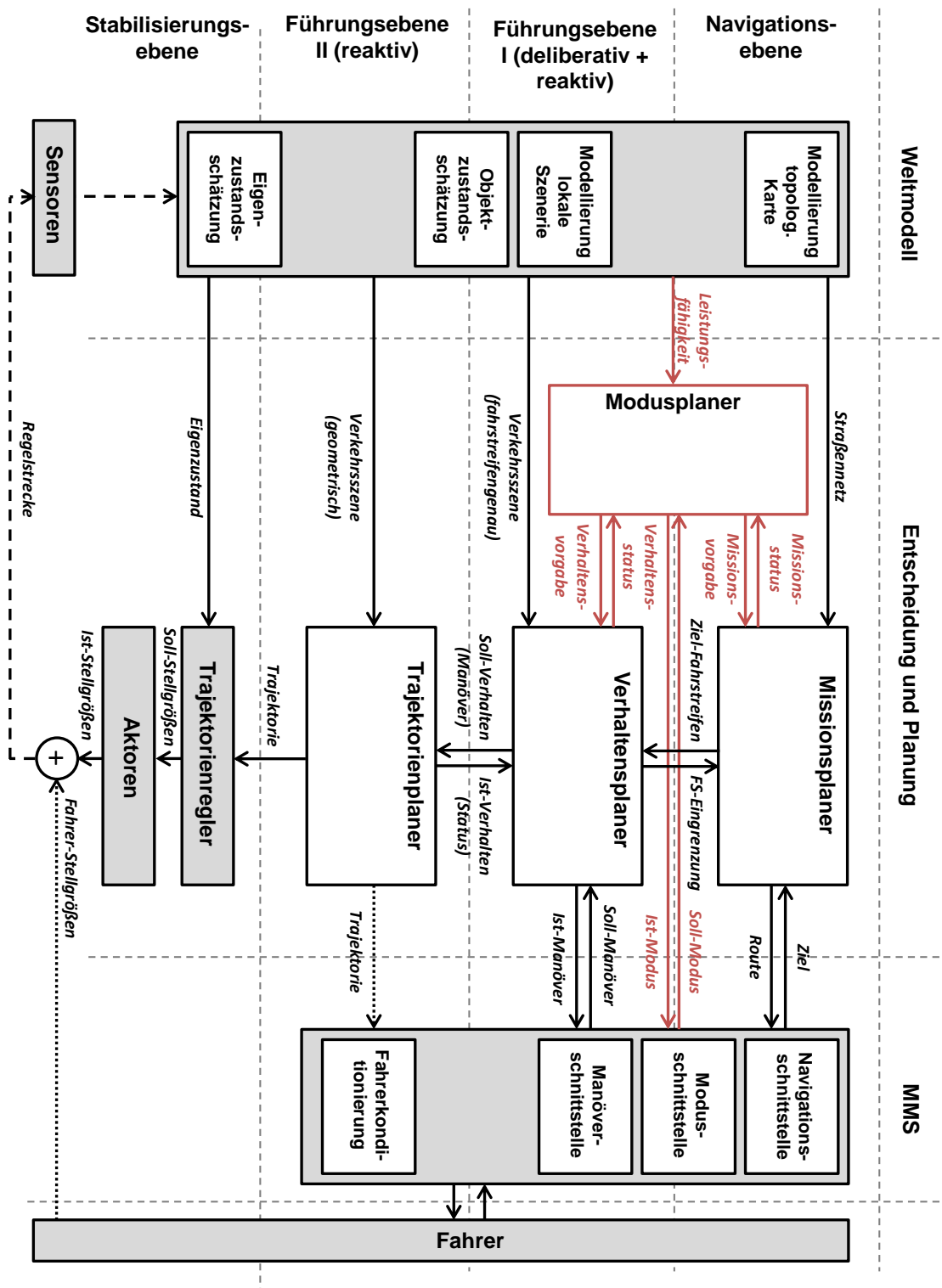


Abbildung 6-1: Oberste Ebene der Bausteinsicht der Referenzarchitektur^{346 347}; Abkürzung: FS: Fahrstreifen

³⁴⁶ Abbildung wurde u.a. auf Basis der folgenden Vorarbeiten erstellt: Lotz (2013): System Architectures for Automated Vehicle Guidance Concepts; Hohm et al. (2014): Automated Driving in Real Traffic

Hinsichtlich der Anbindung des Verhaltensplaners an die MMS ist die Manöverschnittstelle von Bedeutung, die insbesondere für die Umsetzung der Teilautomation auf Manöverebene notwendig ist (s.u. und Abschn. 6.2).

Die dritte Hierarchieebene der Führungsebene II bildet der *Trajektorienplaner*, der die Aufgabe hat, eine konkrete Fahrtrajektorie auf Basis der Manöverborgaben des Verhaltensplaners zu planen. Das Modul nutzt dabei den vom Verhaltensplaner mitgeteilten Entscheidungsspielraum, um eine optimale Trajektorie hinsichtlich definierter Gütekriterien zu finden, die funktionsabhängig, z. B. als Komfortkriterium, gewählt werden können (bspw. durch Minimierung des Rucks als zeitliche Ableitung der Horizontalbeschleunigung)^{348a}. Ein wichtiges Element der Trajektorienplanung ist die Überprüfung der berechneten Trajektorie auf Kollisionsfreiheit^{348a}, der somit einen überwiegenden Teil der Reaktivität des Gesamtsystems abbildet. Um dieser Anforderung gerecht werden zu können, benötigt das Modul eine möglichst exakte geometrische Repräsentation des Umfelds, das Matthaei³⁴⁹ als quasi-kontinuierliche Zustandsgrößen von Umfeld und Eigenzustand bezeichnet. Ebenso spielt dazu die Prädiktion des Verhaltens anderer Verkehrsobjekte auf Trajektorienebene eine wichtige Rolle und muss vom Weltmodell bereitgestellt werden³⁵⁰. Für die Führungsebene II nicht relevant sind abstraktere semantische Informationen, die bspw. Verkehrsregeln repräsentieren. Die Aufteilung der Verantwortung des Systems, ein sicheres (im Sinne von kollisionsfreies) Verhalten zu generieren, auf die beiden Module Verhaltensplaner und Trajektorienplaner kann als strukturelle Redundanz bezeichnet werden (vgl. Kap. 5.2.4). Eine Rückkopplung des Ausführungsstatus des Trajektorienplaners in Bezug auf das vom Verhaltensplaner angeforderte Manöver gilt als Voraussetzung, einen effektiven Kontrollmechanismus im Rahmen einer solchen Redundanz aufrechtzuerhalten.

In der untersten Ebene, der Stabilisierungsebene, findet die Umsetzung und Stabilisierung der angeforderten Trajektorie statt, wobei Werling^{348b} die Regelaufgabe auch als Low-Level-Stabilisierung bezeichnet. Dies entspricht der klassischen Aufteilung nach Donges (vgl. Kap. 5.1.1). Im Gegensatz zu den darüber liegenden Hierarchieebenen lässt die Schnittstelle zwischen Trajektorienplaner und *Trajektorienregler* keinen expliziten Verhaltensspielraum zu, da Abweichungen zwischen Soll- und Ist-Trajektorie letztendlich als Regelabweichung bewertet werden, die es zu minimieren gilt. Der Entscheidungsspielraum für den Trajektorienregler besteht jedoch in der Art und Weise, wie der resultierende notwendige Bewegungsvektor mit Hilfe der zur Verfügung stehenden

³⁴⁷ Als weitere Informations- und Namensgebungsquelle dient auch Matthaei (2015): Wahrnehmungsgestützte Lokalisierung in fahstreifengenaue Karten..., S. 37 ff

³⁴⁸ Werling (2010): Ein neues Konzept für die Trajektoriengenerierung.... a: -; b: S. 12 ff

³⁴⁹ s. Matthaei (2015): Wahrnehmungsgestützte Lokalisierung in fahstreifengenaue Karten..., S. 57

³⁵⁰ vgl. Schreier (2016): Bayesian Environment Representation, Prediction, and Criticality..., S. 156 ff

Aktoren umgesetzt wird³⁵¹. Damit wird auch ersichtlich, dass prinzipiell innerhalb des als Trajektorienregler bezeichneten Moduls eine Arbitrierung mit Assistenzsystemen auf Stabilisierungsebene, wie z. B. des ESC³⁵², stattfinden muss (Letztere sind nicht in Abbildung 6-1 aufgeführt). Durch Umsetzung der Stellgrößen durch die Aktoren und Aufnahme des Eigen- und Umgebungszustandes durch Sensoren wird der Regelkreis des Gesamtsystems schließlich geschlossen.

Eine Sonderrolle und deshalb farblich in Abbildung 6-1 hervorgehoben stellt der *Modusplaner* dar, da er sich auf Basis seiner in Kap. 5.3 beschriebenen Funktion nicht rein einer der hier genannten Architektur-Hierarchieebenen zuordnen lässt. Dennoch bildet der Modusplaner mit den bisher genannten Entscheidungs- und Planungsmodulen ebenfalls eine organisatorische Hierarchie, jedoch nicht in der gleichen Dimension wie das Modell der Fahrzeugführungsaufgabe nach Donges. Da der Modusplaner eine eher administrative und überwachende Rolle in der Architektur einnimmt und die übergeordnete Entscheidung trifft, ob und unter welchen Randbedingungen überhaupt die Fahrzeugführungsaufgabe (teil-) automatisiert ausgeführt wird, könnte das Modul in einer dreidimensionalen Abbildung der Architektur als eine übergeordnete Ebene dargestellt werden, worauf jedoch aus Gründen der Übersichtlichkeit an dieser Stelle verzichtet wird.

Wie bereits in Kap. 5.3 erläutert wurde, stellt der Modusplaner ein Gleichgewicht zwischen dem von außen gewünschtem Betriebsmodus einerseits und situations- bzw. systembedingt zulässigem Betriebsmodus andererseits her. Um den vom Fahrer / Operator gewünschten Betriebsmodus ermitteln und den tatsächlichen Modus an diesen ausgeben zu können, wird eine entsprechende Schnittstelle zur MMS benötigt (s. Abbildung 6-1). Um den situationsbedingt zulässigen Betriebsmodus ermitteln zu können, benötigt der Modusplaner eine Aussage über die Leistungsfähigkeit des Gesamtsystems, die gesammelt vom Weltmodell bereitgestellt wird. Die Schnittstelle beinhaltet neben Kennwerten der Leistungsfähigkeit der Funktionsmodule jedoch auch Kennwerte über die Reichweite und Informationsgüte der Umfeldinformationen, Betriebsmittel und Zustand relevanter Soft- und Hardware-Elemente (Eigendiagnose), insofern sie für die Betriebssicherheit und somit für das Degradationskonzept des Gesamtsystems eine Rolle spielen (vgl. auch Anhang B). Der Modusplaner muss eine Schnittstelle zum Missions- und Verhaltensplaner besitzen, um je nach Betriebsmodus eine automatisiert durchzuführende Fahrmission zu initiieren und / oder direkten Einfluss auf die Verhaltensweisen ausüben zu können.

³⁵¹ vgl. Behere & Törngren (2015): A Functional Architecture for Autonomous Driving

³⁵² Electronic Stability Control, vgl. van Zanten & Kost (2015): Bremsbasierte Assistenzfunktionen

Auch hier spielt der zentrale Gedanke des Entscheidungsspielraums eine übergeordnete Rolle. So ist es denkbar, dass der Modusplaner gezielte Vorgaben bzw. Einschränkungen hinsichtlich der zur Verfügung stehenden Manöver bzw. dessen Ausführungsparameter machen kann (z. B. Herabsenken der max. zulässigen Fahrgeschwindigkeit aufgrund verringerter Sichtweite, Verboten eines Fahrstreifenwechselmanövers aufgrund eines Sensorausfalls etc.), die sich auf die internen Entscheidungsvariablen des Verhaltensplaner-Moduls abbilden. Ebenso ist es denkbar, dass der Modusplaner dem Missionsplaner Einschränkungen hinsichtlich der auswählbaren Straßensegmente zur Routenplanung vorgibt (z. B. das Meiden von nicht-unterstützten oder aktuell nicht bewältigbarer Verkehrsdomänen oder Verkehrsszenen). Die Gefahr, dass in bestimmten Situationen kein Rest-Verhaltensspielraum für Missions- / Verhaltensplaner mehr besteht oder dadurch nicht auflösbare Konflikte zwischen Missions- und Verhaltensplaner entstehen, wird durch die Existenz der rückführenden Schnittstellen (Missions- / Verhaltensstatus) entgegengewirkt, die wiederum zu einer Verhaltensänderung bzw. Änderung des Betriebsmodus seitens des Missionsplaners führen kann.

Die Sub-Module des *Weltmodells*, die die vorher beschriebenen Informationsbedarfe der Entscheidungs- und Planungsmodule stillen, sind in Abbildung 6-1 lediglich grob skizziert, auch da stattfindende Informationsverarbeitungs- und Fusionsprozesse spezifische Implementierungen vorwegnehmen³⁵³. Wie bereits in Kap. 5.4 erwähnt, vertieft bereits Matthaei³⁵⁴ diesen Aspekt der Architekturmodellierung, auf den an dieser Stelle zur weiteren Lektüre verwiesen sei. Vertikal sind die Module so angeordnet, wie sie vorwiegend auf der Abnehmerseite zuzuordnen sind. Eine klare Trennung der jeweiligen Module auf spezifische Ebenen ist nicht immer eindeutig gegeben. So beinhaltet bspw. das Modul *Modellierung lokale Szenerie* bereits sowohl topologische (z. B. Anordnung von Fahrstreifen), semantische (z. B. das Wissen über Verkehrsregeln wie bspw. Geschwindigkeitslimits) und geometrische Informationen (z. B. Verlauf der Fahrstreifenmarkierungen), die u.a. in einer hierarchischen Datenstruktur abgelegt werden können (vgl. Kap. 5.1.2). Dementsprechend wird das genannte Modul in der vorliegenden Referenzarchitektur in beiden Teilen der Führungsebene benötigt, wobei die Planungsmodule auf jeweils unterschiedliche Ebenen der Datenstruktur zugreifen. Ähnlich verhält es sich bspw. mit dem Modul *Eigenzustandsschätzung*, wobei bspw. die Kenntnis der eigenen Horizontalgeschwindigkeit in allen Ebenen der Entscheidungs- und Planungsmodule notwendig ist.

Die Module und Anordnung der Mensch-Maschine-Schnittstelle ergeben sich analog zur Erfüllung der Informationsanforderungen der Planungs- und Entscheidungsmodule.

³⁵³ s. dazu auch Darms (2007): Eine Basis-Systemarchitektur zur Sensordatenfusion...

³⁵⁴ Matthaei (2015): Wahrnehmungsgestützte Lokalisierung in fahrstreifengenauen Karten...

In Abbildung 6-1 sind entsprechende Submodule der MMS definiert. So wird eine *Navigationsschnittstelle* benötigt, die eine Zieleingabe sowie Routenausgabe ähnlich herkömmlicher Navigationssysteme unterstützt. Auf Ebene des Verhaltensplaners besteht je nach Betriebsmodus die Möglichkeit der Vorgabe von Fahrmanövern (vgl. Konzept der *kooperativen Automation* in Kap. 2.1 und 3.4.4), was als *Manöverschnittstelle* bezeichnet wird. Das Sub-Modul *Fahrerkonditionierung* sieht die Möglichkeit vor, dem Fahrer im Falle assistierten Fahrens eine sichere Trajektorie ausgeben zu können (dies wird in Kap. 6.3 vertieft). Die *Modusschnittstelle* als Schnittstelle zum Modusplaner vervollständigt die Submodule der MMS.

An dieser Stelle sei angemerkt, dass in Abbildung 6-1 die Anordnung des Weltmodells auf der linken und der MMS auf der rechten Seite an die RCS-Architektur in Abbildung 3-6 angelehnt ist, was für die Wirkweise der Architektur jedoch unerheblich ist.

6.2 Diskussion

Im Rahmen des vorliegenden Abschnitts werden zur Verifikation der vorgestellten Ergebnisse diese nochmals mit den Anforderungen aus Kap. 2 gegenübergestellt und diskutiert. Insbesondere soll darauf eingegangen werden, wie die Ergebnisse dem Titel der Arbeit gerecht werden, d.h. wie verschiedene Automationsgrade unter Einbeziehung des Fahrers mittels der erarbeiteten Referenzarchitektur umgesetzt werden können.

Im Gegensatz zu Konzepten wie das sog. „H-Mode“, die eine Vielzahl von unterschiedlichen Automatisierungsgraden sowie einen „fließenden“ Übergang zwischen diesen beschreiben³⁵⁵, beschränken sich die Ausführungen dieses Abschnitts auf drei diskret benennbare Automationsgrade: Das vollautomatisierte Fahren als höchste Ausbaustufe des Systems, das assistierte Fahren mit Assistenzfunktionen zur Kollisionsvermeidung sowie einer Zwischenstufe, das teilautomatisierte Fahren auf Manöverebene (vgl. Kap. 1.4).

Abbildung 6-1 zeigt die relevanten Module für die *Vollautomation*, die inhärent alle zur Fahrzeugführung notwendigen Funktionen beinhalten, da sie auf Basis des Modells zur Fahrzeugführung nach Donges erstellt wurden. Die Einbindung des Fahrers beschränkt sich im Wesentlichen auf die Eingabe eines Fahrziels über die Navigationsschnittstelle sowie der Aktivierung bzw. Deaktivierung dieses Modus über die Modusschnittstelle, sobald die vom System unterstützte Verkehrsdomäne erreicht ist bzw. verlassen wird. Eine darüber hinaus gehende Einbindung des Fahrers ist definitionsgemäß bei einem vollautomatisierten Fahrzeug nicht notwendig. Hierbei sei angemerkt, dass im Rahmen

³⁵⁵ vgl. Löper et al. (2008): Kooperative, manöverbasierte Automation...

der vorliegenden Arbeit keine Unterscheidung zwischen den Begriffen „Vollautomatisierung“ und „Hochautomatisierung“ gemäß der Definition nach Gasser et al.³⁵⁶ (oder analog zu „Level 3-“ und „Level 4-Automation“ nach SAE Norm³⁵⁷) erfolgt, da dieser Unterschied nach Sicht des Autors weniger eine technische sondern vielmehr rechtliche Relevanz besitzt.

Im Folgenden sind zwei weitere Aspekte hinsichtlich des vollautomatisierten Modus zu diskutieren. Der erste Aspekt betrifft die Routenplanung: Die Notwendigkeit des Fahrers, ein explizites Fahrtziel einzugeben, um eine Fahrmission zu absolvieren, besteht in der Realität nicht. Die Mission kann auch rein durch Beachtung einer implizit zwischen Fahrer und System abgestimmten Verhaltensregel umgesetzt werden. Eine solche Regel kann bspw. das Folgen eines Autobahn-Segments sein. Ein vollautomatisiertes System sollte demzufolge über die Fähigkeit der *impliziten Missionsplanung* verfügen. Diese Bezeichnung nimmt vorweg, dass eine solche Funktion in der Navigationsebene (Missionsplaner) umgesetzt werden kann. Dies wird im Rahmen der vorliegenden Arbeit empfohlen, da in diesem Fall die hierarchisch darunter liegenden Module keiner Modifikation unterworfen werden müssen, da es für den Verhaltensplaner unerheblich ist, ob eine Navigationsanweisung auf Basis einer expliziten oder impliziten Routenvorgabe erfolgt. Ebenfalls steht dem Missionsplaner eine weitreichendere Vorausschau der Straßensegment-Attribute zur Verfügung, die zum einen für die zuverlässige Verfolgung der impliziten Fahrmission hilfreich ist und zum anderen auf deren Basis eine rechtzeitige Fahrerinteraktion in Form evtl. notwendiger Übergabe-Aufforderungen ausgelöst werden können (Modusplaner). Denkbar sind auch Routen-Präferenzabfragen seitens des Missionsplaners an den Fahrer, falls die implizite Verhaltensregel nicht ausreichend definiert ist, um in allen Situationen eine eindeutige Routenentscheidung herbeizuführen (z. B. bei T-förmigen Verkehrsknotenpunkten). Eine solche Routen-Präferenzabfrage wird der Auflistung der Schnittstelleninformationen in Anhang B hinzugefügt.

Der zweite Aspekt betrifft die Fragestellung, wie im Rahmen der Vollautomatisierung eine Kollisionsvermeidung stattfinden kann und architektonisch eingebunden wird. Dieses Thema wird noch im späteren Verlauf des Abschnitts aufgegriffen.

Die *Teilautomation*, die im Rahmen der Arbeit diskutiert wird, beruht auf dem Konzept der *kooperativen Automation* in der Ausprägung der Manöverdelegation (vgl. Kap. 2.1). Unter Manöverdelegation wird dabei eine zeitdiskrete und fahrerinitiierte Vorgabe auf Manöverebene verstanden, die zu einer neuen oder zur Änderung einer existierenden Bahnführungsentscheidung führt³⁵⁸. Der Modus beinhaltet gemäß Klassifikation nach

³⁵⁶ Gasser et al. (2012): Rechtsfolgen zunehmender Fahrzeugautomatisierung

³⁵⁷ SAE International (2014): Standard J3016 - Taxonomy and Definitions...

³⁵⁸ Definition gemäß Lotz & Winner (2014): Maneuver Delegation and Planning...

Gasser et al.³⁵⁹ eine kontinuierlich automatisierende Funktion in der Längs- und Querverführung, vereint somit u.a. bereits existierende Funktionen wie ACC und Lane Keeping Assistance³⁶⁰ (LKA) und erweitert sie um die Möglichkeit, diskrete Fahrmanöver an das System zu delegieren. Die Umsetzung der Teilautomation benötigt nicht zwingend die Module der Navigationsebene (insbes. den Missionsplaner, s. Abbildung 6-1), da der Fahrer diese Rolle implizit durch Vorgabe von Fahrmanövern über die Manöverschnittstelle einnimmt und eine größere Vorausschau zur rechtzeitigen Rückübergabe der Fahraufgabe an den Fahrer im teilautomatisierten Modus nicht notwendig ist.

Von großer Bedeutung für den objektiven und subjektiven Systemnutzen im teilautomatisierten Modus ist die funktionale Auslegung der Manöverdelegation, die ebenfalls eine Auswirkung auf die Schnittstellen-Definition hat und deswegen näher betrachtet wird. Eine Möglichkeit, das Konzept der Manöverdelegation umzusetzen, ist das Anbieten aktuell möglicher Fahrmanöver an den Fahrer, der diese dann auswählen kann. Im Gegensatz zum Nicht-Anbieten einer solchen „Vorauswahl“ bestehen damit mehrere Vorteile:

- Das System schränkt die Menge der Manöver auf die zur aktuellen Situation passenden ein. Unplausible Manöver müssen daher nicht in unnötigen Dialogen zwischen Verhaltensplaner und Manöverschnittstelle behandelt werden.
- Dies vermeidet zusätzlich Frustrationspotenzial beim Fahrer, dem ansonsten nicht zulässige Manöver nach der Delegation verweigert werden müssten.
- Durch eine geringere Menge an Auswahloptionen verringert sich in vielen Situationen der Bedienungsaufwand sowohl seitens des Fahrers als auch in technischer Hinsicht (Vorhalten der Bedienelemente).

Es verbleibt die Frage, was unter dem Begriff „aktuell mögliche Fahrmanöver“ (s.o.) verstanden wird. Es gibt die Möglichkeit, diese unter Berücksichtigung des statischen Umfelds als auch unter Berücksichtigung des statischen und dynamischen Umfelds anzubieten³⁶¹. In der ersten Variante wird dem Fahrer bspw. ein Fahrstreifenwechsel angeboten, sobald ein Nachbarfahrstreifen vorhanden ist und dieser gemäß den geltenden Verkehrsregeln erreicht werden kann (z. B. keine durchgezogene Fahrstreifenmarkierung). In der zweiten Variante werden zusätzlich die dynamischen Verkehrsobjekte hinsichtlich einer sicheren Ausführung mitberücksichtigt. Im Rahmen dieser Arbeit wird erstere Variante empfohlen, da das System dem Fahrer ansonsten häufig Manöver anbieten und danach wieder verwerfen müsste, was Frustrationspotenzial hat. Vielmehr sollte das System in der Lage sein, die Ausführung des vom Fahrer gewählten Manövers für

³⁵⁹ Gasser et al. (2015): Rahmenbedingungen für die Fahrerassistentenentwicklung, S. 29

³⁶⁰ s. dazu Bartels et al. (2015): Querführungsassistenten

³⁶¹ vgl. hierzu auch die Ausführungen in Winner et al. (2016): PRORETA 3..., S. 1464 ff

einen definierten Zeitraum zu verzögern und ggf. ultimativ zu verwerfen, falls eine sichere Durchführung aufgrund der anderen Verkehrsobjekte nicht zu gewährleisten ist. Eine solche Auslegung der Manöverdelegation schafft eine klare Interaktionsmechanik zwischen Fahrer und System, reduziert die Interaktionen auf ein Minimum und schafft so die Voraussetzung für eine gute Systemakzeptanz (vgl. auch Abschn. 6.3). Die entsprechenden Schnittstelleninformationen werden zur Beschreibung in Anhang B hinzugefügt.

Eine weitere Fragestellung, die das Konzept der Manöverdelegation betrifft, ist das Verhalten des Systems an Kreuzungen³⁶². Eine Variante wäre, dass das System vor jeder Kreuzung eine Manöverdelegation beim Fahrer erzwingt, bevor es die Kreuzung passiert. Dies führt jedoch zu dem Umstand, dass das System häufig und unmotiviert durch äußere Umstände abbremsen müsste. Ebenfalls wäre eine Vielzahl von Interaktionen notwendig, selbst wenn der Fahrer ggf. nur einer innerörtlichen Vorfahrtsstraße folgen möchte. Aus diesem Grund wird, analog zur Fähigkeit der impliziten Missionsplanung, die Notwendigkeit des Systems aufgezeigt, eine *implizite Manöverplanung* zu besitzen (im Verhaltensplaner). Auch hier wird eine implizit zwischen Fahrer und System abgestimmte Verhaltensregel vorausgesetzt (bspw. „Vorfahrtsstraße folgen“). Für eine Vertiefung des Konzepts einer impliziten Manöverplanung (engl.: „default maneuver“) wird an dieser Stelle auf Lotz & Winner³⁶³ verwiesen.

Auf Basis der erarbeiteten Referenzarchitektur ist eine Kombination des vollautomatisierten Modus mit dem Konzept der Manöverdelegation denkbar: So erlaubt die Architektur im vollautomatisierten Modus eine zusätzliche Manöverdelegation des Fahrers. Eine solche Auslegung besitzt den Vorteil, dass der Fahrer, sofern er es für notwendig erachtet, trotz Vollautomation in die Bahnführungsentscheidung eingreifen kann und somit das Systemverhalten gemäß eigener Präferenz mit beeinflussen kann. Dies schafft einen Mehrwert ohne signifikante zusätzliche Kosten hinsichtlich Architektur und Implementierung. Einzig die Implementierung der Arbitrierung des Fahrerwunsches innerhalb des Verhaltensplaners könnte einen Mehraufwand bedeuten, sofern dem Fahrer die Möglichkeit gegeben werden soll, durch eine Manöverdelegation indirekt die Routenführung zu ändern.

Zuletzt wird die Umsetzung des *Assistenzmodus* diskutiert. Hierbei steht jedoch, wie bereits im vorherigen Teil des Abschnitts erwähnt, die Funktion der Kollisionsvermeidung bzw. Kollisionsfolgenminderung im Vordergrund (Kategorie der *eingreifenden Notfallfunktionen* gemäß Gasser et al.³⁶⁴). Zur Umsetzung dieses Modus sind gemäß der

³⁶² vgl. auch Geyer (2013): Entwicklung und Evaluierung eines kooperativen Interaktionskonzepts...

³⁶³ Lotz & Winner (2014): Maneuver Delegation and Planning...

³⁶⁴ Gasser et al. (2015): Rahmenbedingungen für die Fahrerassistentenentwicklung, S. 29

vorgestellten Referenzarchitektur analog zur Teilautomation alle Module gemäß Abbildung 6-1 notwendig mit Ausnahme der Module der Navigationsebene. Dass eine eingreifende Notfunktion nicht auf die Existenz einer explizit oder implizit vorgegebenen Fahrmission in Form einer Fahrtroute angewiesen ist oder sein sollte, wird vorausgesetzt³⁶⁵. Zur Beantwortung der Frage, wie eine Funktion zur Kollisionsvermeidung in der Referenzarchitektur abgebildet wird, sei auf die Grundsätze hybrider Basisarchitekturen verwiesen (s. Kap. 3.3). Dort werden auf hierarchisch niedriger Ebene die reaktiven Verhaltensweisen des Systems implementiert. Da eine Funktion zur Kollisionsvermeidung rein reaktiver und nicht deliberativer Natur ist, muss ein solches Verhalten demnach ebenfalls auf einer reaktiven Systemebene in der erarbeiteten Referenzarchitektur abgebildet werden. Im Hinblick auf Abbildung 6-1 wird deswegen empfohlen, ein solches Verhalten im Trajektorienplaner abzubilden. Demnach wird konzeptuell die Kollisionsvermeidung als ein weiteres, eigenes Verhalten dem Repertoire des Trajektorienplaners (neben den in Tabelle 2-1 aufgezählten Fahrmanövern) hinzugefügt. Dies ist auch deswegen mit der Referenzarchitektur vereinbar, da insbesondere auf der Führungsebene II quasi-kontinuierliche, geometrische Informationen des Umfelds verwendet werden, also Informationen, die für eine solche Funktion notwendig sind. Semantische oder topologische Weltmodell-Informationen sind in diesem Kontext von geringerer Bedeutung.

Dennoch spielen zur Vervollständigung der Funktion der Verhaltens- und Modusplaner eine weitere Rolle, die sich vor allem aus dem Zusammenspiel mit den anderen Betriebsmodi begründet. Da auch in den anderen beiden Modi eine sichere Fahrzeugführung einschließlich der Fähigkeit zur Kollisionsvermeidung notwendig ist, muss das Verhalten der Kollisionsvermeidung zur Laufzeit mit den anderen Verhalten der automatisierten Fahrfunktionen arbitriert werden. Dies begründet sich dadurch, dass eine inhärente Kopplung einer Kollisionsvermeidung mit Automationsfunktionen in der Trajektorienplanung (d.h. als eine Eigenschaft der Verhaltensweisen) zwar denkbar ist, jedoch konzeptuelle Nachteile mit sich bringt, da zu jeder Zeit alle hochdynamischen Fahrdynamik-Anforderungen in evtl. vorhandenen Fahrdynamikmodellen der Entscheidungsmodule berücksichtigt werden müssen, was den Umsetzungsaufwand erheblich steigert. Ebenfalls werden zur Ableitung fahrstrategischer Verhaltensentscheidungen im Verhaltensplaner ggf. andere Prädiktionsmodelle zur Vorhersage der Bewegung der anderen Verkehrsobjekte genutzt, als sie für eine Kollisionsvermeidung zulässig sind. Dies bedeutet schließlich, dass die Entscheidung, zu welchem Zeitpunkt eine „reguläres“ Verhalten zur Automation oder das Verhalten zur Kollisionsvermeidung ausgeführt wird, im Verhaltensplaner getroffen werden muss.

³⁶⁵ Vielmehr kann umgekehrt formuliert werden, dass eine Fahrmission jedweder Art eine sichere Ausführung dieser einschl. einer Fähigkeit zur Kollisionsvermeidung voraussetzt.

Dazu wird im Rahmen der vorliegenden Arbeit empfohlen, dass das Verhalten der Kollisionsvermeidung (implementiert im Trajektorienplaner) einen möglichst funktionsunabhängigen und allgemeingültigen Kennwert an den Verhaltensplaner rückkoppelt, der die Kritikalität des Verhaltens ausdrückt. Auf Basis eines solchen Kennwertes kann der Verhaltensplaner dann entscheiden, ob und zu welchem Zeitpunkt die Kollisionsvermeidung aktiviert wird. Eine derartige Rückkopplung bietet den Vorteil, dass die Aufschaltung des Kollisionsvermeidungsverhaltens nicht nach einer starren Regel erfolgt, sondern unter Berücksichtigung situativer Aspekte variiert werden kann:

- Der Zeitpunkt der Aktivierung der Kollisionsvermeidung kann auf die aktuelle Fahrer-Aufmerksamkeit angepasst werden (z. B. zeitlich spätere Aktivierung bei aufmerksamen Fahrer im assistierten und teilautomatisierten Modus), um die Wahrscheinlichkeit von falsch-positiven Auslösungen zu reduzieren.
- Der Zeitpunkt der Aktivierung der Kollisionsvermeidung kann auf die aktuelle Gesamtleistungsfähigkeit des Systems angepasst werden. So kann über die Funktionskette des Modusplaners und der Schnittstelle der Verhaltensvorgabe (vgl. Abbildung 6-1) bspw. auf Situationen widriger Witterungsbedingungen reagiert werden.
- Der Zeitpunkt der Aktivierung der Kollisionsvermeidung kann auf die aktuelle Leistungsfähigkeit der geplanten Fahrmanöver angepasst werden. So kann das System eine Aktivierung der Kollisionsvermeidung verzögern, falls bspw. aufgrund der aktuellen Verkehrssituation ein etwas dynamischeres bzw. riskanteres Fahrmanöver ausgeführt werden muss (bspw. Fahrstreifenwechsel unter Zwang).

Da der Eingriffszeitpunkt des Systems variabel ist und im Verhaltensplaner entschieden wird, müssen im assistierten Modus auch Fahrerwarnungen und Verhaltensempfehlungen auf dieser Ebene der Schnittstelle zur MMS hinzugefügt werden (vgl. Anhang B). Eine beispielhafte Umsetzung der Kollisionsvermeidung als Assistenzfunktion einschließlich der o.g. Eingriffsentscheidung wird später in Kap. 6.3 aufgezeigt.

Abschließend werden im Rahmen des vorliegenden Abschnittes nochmals die in Kap. 2.2.3 aufgeführten funktionalen Anforderungen rekapituliert. Eine Einordnung, wie die notwendigen Verhaltensweisen aus Tabelle 2-1 in der erarbeiteten Referenzarchitektur abgebildet werden, ist erfolgt (vgl. insbes. Kap. 5.2.4). Darüber hinaus wurden die weiteren Anforderungen ebenfalls adressiert:

- **Fähigkeit zur Missionsplanung und -ausführung:** Wird im Missionsplanungsmodul abgebildet.
- **Fähigkeit zur Verhaltensausführung:** Wird in der Führungsebene II sowie der Stabilisierungsebene abgebildet.
- **Assistiertes (unfallvermeidendes) Fahren:** Wie im vorliegenden Abschnitt erläutert.

- **Mensch-Maschine-Schnittstelle:** Durch explizite Aufnahme eines eigenen Moduls (MMS) in der Referenzarchitektur.

Anforderungen, die die erarbeitete Referenzarchitektur nicht erfüllen kann sind:

- Navigation in stark unstrukturierten Umgebungen
- Das Planen und Ausführen von Missionselementen, die nicht zur Verfolgung einer Fahrtroute gehören (z. B. die Steuerung von Anbaugeräten, wie sie in Nutzfahrzeugen zu finden sind)³⁶⁶
- Das Berücksichtigen von mehr Automationsgraden als sie im vorliegenden Abschnitt aufgeführt wurden

Ebenfalls in der Architekturdiskussion nicht weiter vertieft wurden Anforderungen, die sich aus der funktionalen Sicherheit der automatisierten Fahrzeugführung ergeben, wie z. B. Maßnahmen der Aufrechterhaltung der Funktionalität im Fehler- / Versagensfall sowie möglichen Redundanzkonzepten von Hard- und Softwarekomponenten. Hierbei wird vermutet, dass eine tiefergehende Definition des Modusplaners und der damit verbundenen Überwachungs- und Eigendiagnosefunktionen des Gesamtsystems notwendig ist.

6.3 Umsetzung im Projekt PRORETA 3

Im vorliegenden Abschnitt wird eine Umsetzung der erarbeiteten Referenzarchitektur im Rahmen des Forschungsprojektes PRORETA 3 aufgezeigt. Eine Beschreibung und Motivation des Projektes ist bereits in Kap. 2.1 erfolgt. Teile des Abschnittes basieren darüber hinaus auf Winner et al.³⁶⁷.

Abbildung 6-2 zeigt neben den am Projekt beteiligten wissenschaftlichen Mitarbeitern das Versuchsfahrzeug, das zur Entwicklung und Verifikation der PRORETA 3-Funktionen von der Continental AG zur Verfügung gestellt wurde. Beim Fahrzeug handelt es sich um einen modifizierten Volkswagen Passat (B7). Das Fahrzeug verfügt über Schnittstellen zur Beeinflussung der Längs- und Querdynamik (EPS-Lenkmomenten-Schnittstelle³⁶⁸; Geschwindigkeits- und Beschleunigungsanforderung über eine interne ACC-Schnittstelle sowie ESC). Sensorseitig verfügt das Fahrzeug über einen Fernbereichs-Radarsensor im Frontbereich, vier Nahbereichs-Radarsensoren (je zwei im vorderen und hinteren Bereich des Fahrzeugs) sowie einer Stereokamera, wie in Abbildung

³⁶⁶ s. z. B. Reinards et al. (2015) Fahrerassistenzsysteme bei Traktoren

³⁶⁷ Winner et al. (2016): PRORETA 3: Comprehensive Driver Assistance...

³⁶⁸ EPS: Electric Power Steering; Elektromechanisches Lenksystem

6-3 dargestellt. Darüber hinaus verfügt das System über einen Verbund aus mehreren PCs, auf denen neben den Systemfunktionen auch eine Messtechnik-Schnittstelle integriert ist. Die Datenkommunikation zwischen den Modulen wird über eine Ethernet-basierte Schnittstelle mit sog. Publisher-Subscriber-Topologie umgesetzt.



Abbildung 6-2: Das PRORETA 3 Versuchsfahrzeug mit wissenschaftlichen Mitarbeitern Matthias Schreier, Matthias Pfromm, Felix Lotz, Eric Bauer (v.l.n.r.)

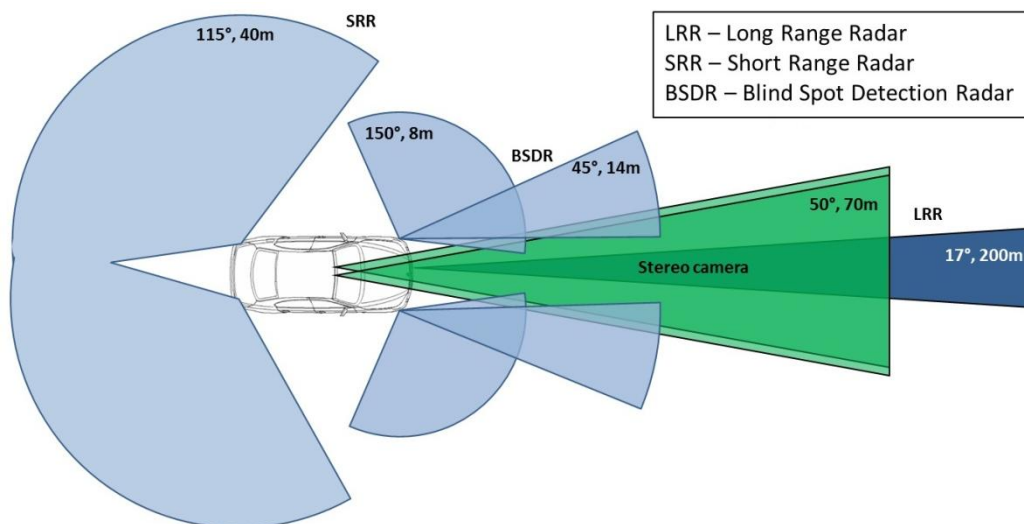


Abbildung 6-3: Schematische Darstellung der Sensor-Infrastruktur des PRORETA 3 Versuchsfahrzeugs

Die Ergebnisse der vorliegenden Arbeit basieren maßgeblich auf den Erkenntnissen, die im Zuge der Erstellung der funktionalen Architektur für das PRORETA-System gewonnen wurden. Die umgesetzte Architektur von PRORETA 3³⁶⁹ kommt ohne die in

³⁶⁹ vgl. Winner et al. (2016): PRORETA 3: Comprehensive Driver Assistance..., S. 1452 ff

Abbildung 6-1 dargestellte Navigationsebene aus, da im Rahmen des Projekts keine Vollautomatisierung umgesetzt wurde. Eine Abweichung von der Referenzarchitektur ist darüber hinaus die algorithmische Zusammenlegung des Modusplaners mit dem Verhaltensplaner, da im PRORETA-Projekt vereinfachend von einer konstanten Leistungsfähigkeit des Systems ausgegangen wurde und auf eine komplexe Überwachungsfunktion aufgrund des reinen Einsatzes auf einem abgesperrten Versuchsgelände verzichtet wurde.

Der kombinierte Modus- und Verhaltensplaner des PRORETA-Systems wurde als hierarchischer Zustandsautomat implementiert, in dem der Modusplaner die oberste Hierarchie-Ebene einnimmt, siehe Abbildung 6-4.

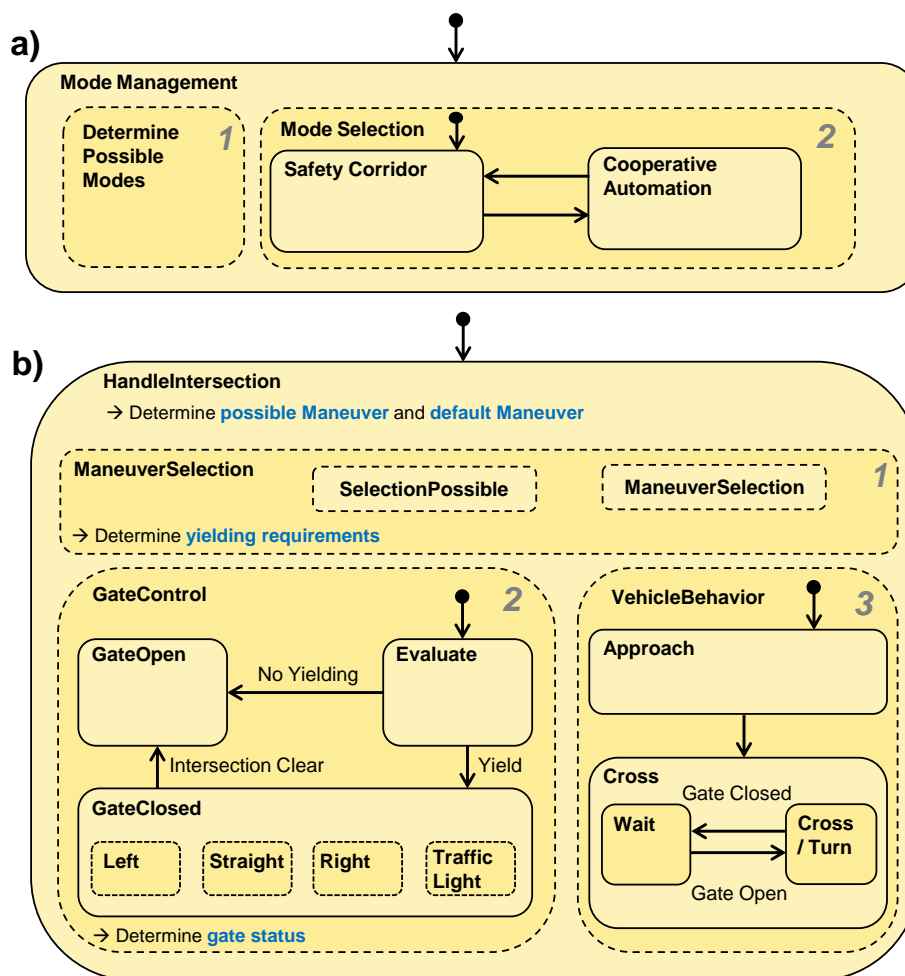


Abbildung 6-4: a) Vereinfachtes Zustandsdiagramm des PRORETA 3 Modusplaners; b) Vereinfachtes Zustandsdiagramm des Verhaltensplaners im Unterzustand „Cooperative Automation“ und „Handle Intersection“; gestrichelte Boxen zeigen nebenläufige (parallele) Zustände und durgezogene Boxen sequentielle Zustände an

Dabei wird in Abbildungsteil a) zunächst der nebenläufige Zustand „Determine Possible Modes“ ausgeführt, der die wichtigsten Vorbedingungen prüft, die zum Anbieten der Modi notwendig sind (z. B. die Verfügbarkeit der beteiligten Softwaremodule). Diese

Bedingungen werden genutzt, um in Kombination mit einem möglichen Modus-Wechselwunsch des Fahrer zwischen den beiden Automationsmodi *Sicherheitskorridor* und *kooperative Automation* zu wechseln (rechter Teil von Abbildung 6-4 a)) .

Eine beispielhafte Implementierung des Verhaltensplaners im Modus *kooperative Automation* für die Situation der Bewältigung von zweistreifigen Kreuzungen (je Zweig) zeigt Abbildung 6-4 b). Dabei werden im ersten Schritt einmalig die an der Kreuzung durchführbaren Fahrmanöver sowie das „default maneuver“ ermittelt (Folgen der Vorfahrtsstraße, vgl. Abschn. 6.2). Anschließend erfolgt im ersten von insgesamt drei nebenläufigen Zuständen („ManeuverSelection“) die Manöverauswahl einschließlich der Berücksichtigung des vom Fahrer delegierten Manövers. Erst auf Basis dieser Information und unter Kenntnis der gültigen Verkehrsregelung ist eine Aussage möglich, welche Vorfahrtsregel anzuwenden ist („determine yielding requirements“). Im zweiten Zustand („GateControl“) wird auf Basis dieser Information ermittelt, ob ein Einfahren in die Kreuzung möglich ist oder die Kreuzung in Analogie zum Gate-Konzept nach Geyer³⁷⁰ „verschlossen“ bleibt. Im dritten und letzten nebenläufigen Zustand („VehicleBehavior“) wird letztendlich das aktuell auszuführende Fahrmanöver ermittelt. So ist, solange sich das Fahrzeug noch vor dem Einfahrtspunkt in die Kreuzung befindet, das Verhalten „Annähern an Knoten“ (im Bild: „Approach“) aktiv. Erst wenn das Fahrzeug den Einfahrtspunkt erreicht hat und alle Vorfahrtsbedingungen erfüllt sind (Unterzustand „GateClosed“ geht über auf „GateOpen“) und somit ein gefahrloses und verkehrsregelkonformes Kreuzen bzw. Abbiegen möglich ist, wird das entsprechende Manöver an die Trajektorienplanung beauftragt.

Dieses Beispiel zeigt auch, dass für die Verhaltensplanung eine topologische und semantische Szenenmodellierung notwendig ist, um bspw. die Kreuzungstopologie und geltende Vorfahrtsregelungen korrekt ermitteln zu können. Im PRORETA Projekt wurde dazu das Versuchsgelände des August-Euler-Flugplatzes in Griesheim mittels des OpenStreetMap-Standards³⁷¹ kartographiert und die Karte um eigene semantische und topologische Attribute erweitert. Basierend auf der Arbeit von Schönichen³⁷² wurde ein „elektronischer Horizont“ implementiert, der schließlich die Karteninformationen auf Basis der kartenreferenzierten Fahrzeugposition in strukturierter Art und Weise zur Verfügung stellt. Für eine tiefergehende Betrachtung sei an dieser Stelle auf Schönichen verwiesen.

³⁷⁰ Geyer (2013): Entwicklung und Evaluierung eines kooperativen Interaktionskonzepts...

³⁷¹ siehe <https://www.openstreetmap.org>, letzter Zugriff am 03.10.16

³⁷² Schönichen (2012): Integration von Informationen digitaler Karten in ein Fusions-Framework..., S. 28

Neben dem elektronischen Horizont besteht die lokale Szenenmodellierung aus einer Schätzung der fahrzeugrelativen Fahrstreifenverläufe auf Basis der o.g. Stereokamera sowie einer Objektliste aus den fusionierten Radar- und Kamera-Objekthypothesen.

Auf der Führungsebene I des PRORETA-Systems ist schließlich noch die Modus- und Manöverschnittstelle implementiert. Abbildung 6-5 zeigt die dazugehörigen Bedienelemente als modifizierte Lenkstockhebel von ursprünglich ACC- und Fahrtrichtungsanzeiger-Funktion. In Abbildung 6-6 ist ein Teil des Anzeigekonzepts der Manöverschnittstelle dargestellt, wobei im sog. frei programmierbaren Kombiinstrument (FPK) aktuell ausgeführte Fahrmanöver als ausgefüllter Pfeil (im Bild: An Kreuzung annähern) und aktuell auswählbare Manöver als leerer Pfeil dargestellt werden (im Bild: Links- bzw. Rechts-Abbiegen an T-förmiger Kreuzung). Für eine Vertiefung zum MMS-Konzept sei an dieser Stelle auf Pfromm³⁷³ verwiesen.



Abbildung 6-5: PRORETA 3-Bedienelemente zur Modusschnittstelle (links) und Manöverschnittstelle (rechts)



Abbildung 6-6: Anzeigekonzept zur Manöverschnittstelle

³⁷³ Pfromm (2016): Menschbezogener Umgang mit Systemstörungen...

Auf der Führungsebene II des PRORETA-Systems versucht die Trajektorienplanung, unabhängig vom Automationsmodus des Systems, einen durchgängigen algorithmischen Ansatz umzusetzen und somit eine einheitliche Lösung in beiden Systemmodi zu implementieren. Das Modul wird als Potenzialfeld-basierter Ansatz formuliert, in dem statische Hindernisse, dynamische Objekte und Fahrstreifenmarkierungen je ein zweidimensionales Potenzialfeld erzeugen, die dann additiv zu einem resultierenden Feld zusammengefasst werden. Idealisiert betrachtet stellt das aufsummierte Potenzialfeld ein „Gefahrengebirge“ dar und das Tal darin den sichersten Pfad, den das System verfolgen sollte. Da die resultierende Trajektorie jedoch noch den Randbedingungen unterliegt, dass Kinematik und Fahrdynamik des Fahrzeugs sowie Stellgrößenbeschränkungen bei der Lösungsfindung berücksichtigt werden müssen, wird die Planung als ein Optimalsteuerungsproblem (genauer: nichtlineares, modellprädiktives Regelungsproblem - NMPC³⁷⁴) definiert. Für eine vertiefende Betrachtung zur Trajektorienplanung sei an dieser Stelle auf Bauer & Konigorski³⁷⁵ verwiesen.

Der Trajektorienplanungsansatz kann auch im Modus *kooperative Automation* genutzt werden. Dort führt eine Manöverdelegation zu einer Anpassung des Potenzialfelds, sodass auch Fahrmanöver umgesetzt werden können³⁷⁶ (z. B. Absenkung des Potenzials auf einem Nachbar-Fahrstreifen, sobald ein Fahrstreifenwechsel beauftragt wird).

Als weitere notwendige Eingangsgröße für den Trajektorienplaner auf der Führungsebene des Systems muss, wie bereits zuvor erwähnt, eine Repräsentation des statischen Umfelds vom Weltmodell bereitgestellt werden. Hierzu wurde eine sog. parametrische Freiraumkarte von Schreier^{377a} entwickelt und implementiert, die auf einer 2D-Belegungskarte auf Radar- und Kamerabasis erstellt wird³⁷⁸. Ergebnis des Algorithmus ist eine B-Spline-basierte Repräsentation der Frei- und Belegtbereiche um das Fahrzeug. Darüber hinaus benötigt die Trajektorienplanung zur Berücksichtigung der Kollisionsfreiheit eine Prädiktion der anderen Verkehrsobjekte für zukünftige Zeitpunkte, die als weiteres Element dem Modul der Objektzustandsschätzung hinzugefügt wird. Für eine vertiefende Betrachtung der Freiraumkarte sowie der Objektprädiktion sei an dieser Stelle auf Schreier verwiesen^{377b}.

Die Umsetzung der Eingriffsentscheidung im Modus *Sicherheitskorridor* erfolgt im PRORETA-System wie bereits in Abschn. 6.2 beschrieben: Die Kollisionsvermeidung ist als eigenes Verhalten im Trajektorienplaner implementiert. In jedem Berechnungs-

³⁷⁴ Non-linear model-predictive control

³⁷⁵ Bauer & Konigorski (2013): Ein modellprädiktiver Querplanungsansatz zur Kollisionsvermeidung

³⁷⁶ vgl. Winner et al. (2016):); PRORETA 3: Comprehensive Driver Assistance..., S. 1461

³⁷⁷ Schreier (2016): Bayesian Environment Representation, Prediction, and Criticality.... a: S. 103 ff; b: -

³⁷⁸ vgl. Grewe (2014): Optimierung der Repräsentation von Occupancy-Grids...

schritt wird die resultierende, optimale Trajektorie hinsichtlich des Maximums (des Betrags) der damit einhergehenden Horizontalbeschleunigung durchsucht. Dieser Wert wird ständig an den Verhaltensplaner geschickt, der somit einen Anhaltspunkt zur Bewertung der Kritikalität besitzt, in dem sich das System befindet. Ausschlaggebend für die Eingriffsentscheidung ist hierbei der Ausnutzungsgrad des zur Verfügung stehenden Reibbeiwerts zwischen Reifen und Fahrbahn, sodass auch den aktuellen Witterungsbedingungen Rechnung getragen wird (bspw. würde bei einer Reibbeiwert-Ausnutzung von 80 % eingegriffen). Im PRORETA-System wird die Eingriffsentscheidung zusätzlich von der Blickrichtung des Fahrers abhängig gemacht. Blickt der Fahrer nach vorne, erfolgt die Eingriffsentscheidung später, scheint der Fahrer abgelenkt, früher. Hierzu ist im Versuchsfahrzeug eine Fahrerbeobachtungskamera verbaut, die die Kopfpose schätzt³⁷⁹.



Abbildung 6-7: Visualisierung des Sicherheitskorridors als Element der Fahrerkonditionierung im FPK (siehe Verformung des Tachometer-Kreises); im Bildschirm rechts: Darstellung der 2D-Belegungskarte

PRORETA 3 nutzt die Schnittstelle der Fahrerkonditionierung im Modus *Sicherheitskorridor*. Dieses Modul bestimmt auf Basis der übermittelten sicheren Trajektorie eine abstrahierte räumliche Darstellung möglicher Gefahrenstellen und visualisiert diese für den Fahrer, wie am Beispiel einer Engstelle in einer Baustellensituation in Abbildung 6-7 dargestellt ist.

³⁷⁹ vgl. Pfromm et al. (2013): Driver Assistance via Optical Information with Spatial Reference

Die folgenden Szenarien wurden im Projekt umgesetzt:

Szenarien zum Sicherheitskorridor:

- Automatischer Bremseneingriff auf ein plötzlich erscheinendes Hindernis
- Automatischer Bremseneingriff auf ein statisches Hindernis
- Eingriff bei überhöhter Kurvengeschwindigkeit
- Kollisionsvermeidung in einer engen Baustelle
- Eingriff bei unbeabsichtigtem Verlassen des Fahrstreifens
- Verhindern des Überfahrens des Verkehrszeichens „Einfahrt verboten“ als Beispiel für einen Verkehrsregel-basierten Eingriff
- Verhindern des Überfahrens von Lichtsignalanlagen mit roter Phase als Beispiel für einen Verkehrsregel-basierten Eingriff

Szenarien zur kooperativen Automation:

- Automatisches Fahrstreifenwechsel-Manöver
- Fahrstreifenreduktion: Automatisches Anhalten bei einem endenden Fahrstreifen, falls kein Fahrstreifenwechsel beauftragt wird
- Verkehrszeichen-geregelte Kreuzung einschließlich „default maneuver“-Konzept, Anbieten der Abbiegemanöver und automatischem Abbiegen
- Automatisiertes Fahren durch eine Baustelle
- Lichtsignalanlagen-geregelte Kreuzung einschließlich „default maneuver“-Konzept, Anbieten der Abbiegemanöver und automatischem Abbiegen unter Berücksichtigung der Lichtsignalanlagen-Phase

Das PRORETA 3-System einschließlich der o.g. Szenarien wurde am 11. und 12. September 2014 auf dem August-Euler Flugplatz in Griesheim im Rahmen einer Abschlussveranstaltung der Presse und Gästen aus Industrie und Forschung präsentiert. Eine Zusammenfassung der Veranstaltung einschließlich Fachpräsentationen sowie einem Demo-Video sind auf der PRORETA-Homepage zu finden³⁸⁰.

Die Ausführungen dieses Abschnittes geben einen Einblick, wie Teile der erarbeiteten Referenzarchitektur konkret implementiert werden können, sodass sich für den Fahrer ein konsistenter Gesamteindruck des Systems mit hohem Nutzwert ergibt³⁸¹. Trotz signifikanter Vereinfachungen, die im Rahmen der Implementierung durch Schwerpunktsetzungen des Projektes gemacht werden mussten (inbes. bei der Berücksichtigung dyn. Verkehrsobjekte und durch die Entwicklung auf einem abgesperrten Gelände), zeigt das Ergebnis des Projektes, dass die erarbeitete Referenzarchitektur

³⁸⁰ <http://www.proreta.tu-darmstadt.de/proreta/proreta3>, letzter Zugriff am 03.10.16

³⁸¹ Ergebnis einer Kurz-Studie im Rahmen der o.g. Abschlussveranstaltung, vgl. Winner et al. (2016): PRORETA 3: Comprehensive Driver Assistance..., S. 1465 ff

insbesondere hinsichtlich des Zusammenspiels der Führungs- und Stabilisierungsebenen in der Lage ist, die beiden Systemmodi assistiertes und teilautomatisiertes Fahren in einem konsistenten Gesamtansatz umzusetzen.

Eine Verifikation der Erreichung der nichtfunktionalen Anforderungen und somit der Gesamtqualität des Architekturentwurfs ist mit der geschilderten Umsetzung jedoch nur eingeschränkt möglich, aber für folgende Kriterien hat sich der Architekturentwurf bewährt:

- Der Architekturentwurf musste während der Projektlaufzeit nicht geändert werden und hielt einer Vielzahl von Diskussionen im Projektkonsortium stand.
- Der Architekturentwurf konnte die Vielzahl der Systemfunktionen (siehe o.g. Szenarien) abbilden, ohne dass nachträgliche „Anbauten“ oder zusätzliche Schnittstellen hinzugefügt werden mussten (obwohl die o.g. Szenarien zum Zeitpunkt des Architekturentwurfs noch nicht festgelegt waren).
- Der Architekturentwurf ermöglichte eine gute Arbeitsteilung im Projektteam, was für einen hohen Grad an semantischer Kohäsion spricht.
- Die implementierten Module konnten aus der PRORETA-Architektur ausgelöst und dem Projektpartner Continental übergeben werden, wobei diese teilweise in einem anderen Funktionskontext genutzt werden konnten (Modul zur Berechnung der parametrische Freiraumkarte als Teil der lokalen Szenenmodellierung sowie Trajektorienplaner). Dies spricht für einen hohen Grad an Modularität.

Einen praktischen und „alltäglichen“ Nutzen des Architekturentwurfs konnte durch eine Modellierung der Architektur in UML-Notation erreicht werden, einschließlich einer Auflistung der konkreten Modulfunktionen und Schnittstellen. Im Rahmen der Entwicklungsarbeiten diente diese detaillierte Darstellung, die im Projektbüro als Plakat ausgeführt wurde, als geeignete Diskussionsgrundlage und Hilfestellung zur Lösungsfindung in Implementierungsfragen.

7 Fazit und Ausblick

Zielsetzung der vorliegenden Arbeit ist es, eine Referenzarchitektur für assistiertes und automatisiertes Fahren zu entwickeln, wobei insbesondere erstmals der Erkenntnisweg, der zum letztendlichen Ergebnis führt, explizit aufgezeigt werden soll.

Eine Grundlagenrecherche zur Architekturentwicklung macht deutlich, dass die Qualität eines Architekturentwurfs nicht durch die Erfüllung funktionaler Anforderungen definiert ist, sondern vielmehr durch die der nichtfunktionalen Anforderungen. Gleichzeitig existiert jedoch zum Zeitpunkt des Architekturentwurfs keine Metrik, die eine quantifizierbare Bewertung des Erfüllungsgrades nichtfunktionaler Anforderungen zulässt. Um dieses Dilemma zu überwinden, wird im Rahmen der vorliegenden Arbeit eine Architektur-Entwurfsmethodik entwickelt, die das Problem als ein heuristisches Entscheidungsverfahren formuliert und die Problemfaktorisierung als ein Kernelement zur Lösungsfindung einsetzt. Dabei wird das Gesamtproblem in sequenziell lösbare Teilprobleme zerlegt, wobei eine schrittweise Entwicklung der Architektur vom Grob- bis zur Feinentwurf erfolgt. Für jeden Konkretisierungsschritt werden eine möglichst den Lösungsraum vollständig abdeckende Anzahl an Entwurfsvarianten gebildet und anhand von Entscheidungskriterien, die Rückschlüsse auf die Erfüllung für die im Rahmen dieser Arbeit ausgewählten nichtfunktionalen Anforderungen geben, auf qualitativer Basis gegeneinander bewertet. Dieser Vorgang macht den Entscheidungsprozess, der hinter einer Architekturentwicklung steht, sichtbar und nachvollziehbar.

Die entwickelte Methodik konnte erfolgreich auf die im Rahmen der vorliegenden Arbeit betrachtete Domäne der A/A-Systeme angewendet werden. Als Ergebnis liegt eine Referenzarchitektur für assistiertes und automatisiertes Fahren vor, die als Vertreter hybrider Roboter-Basisarchitekturen die Vorteile reaktiver und deliberativer Systeme vereint. Als Vorbild zur Gliederung der Hierarchieebenen der Architektur dient dabei das Modell zur Fahrzeugführung nach Donges, wobei insbesondere die Führungsebene einen Großteil der Komplexität automatisierter Fahrzeugführungssysteme beinhaltet. Aus diesem Grunde erfolgt eine Zweiteilung der Führungsebene in die Module Verhaltensplaner und Trajektorienplaner, die gemeinsam die Reaktivität des Gesamtsystems sicherstellen. Zentrales Element der Architektur ist dabei der Entscheidungsspielraum, über den die einzelnen Planungsmodule verfügen und der im Rahmen der Arbeit explizit für jedes Modul benannt ist. Mit Hilfe dieses Konzeptes wird auch das Problem der Kombination deliberativer und reaktiver Systemelemente gelöst, wobei der Verhaltensplaner grundsätzlich frei über die durchzuführenden Fahrmanöver entscheiden kann, sofern eine Erreichbarkeit der Zielfahrstreifen vor und nach Knotenpunkten gegeben ist. Auch komplexe Fahrscenarien, in denen eine Manöverentscheidung an eine Erwartung

an deren Ausführung gekoppelt ist, können durch Vorgabe des Verhaltensspielraums von Verhaltens- an Trajektorienplaner umgesetzt werden. Die Referenzarchitektur ist in Form einer Bausteinsicht (s. Abbildung 6-1) sowie einer Schnittstellenbeschreibung (s. Anhang B) implementierungsunabhängig dokumentiert.

Die Umsetzung der Architektur im Rahmen des Forschungsprojektes PRORETA 3 zeigt zum einen Möglichkeiten zur konkreten Implementierung eines Teils der in der Referenzarchitektur genannten Module auf und verifiziert zum anderen die Erfüllung der funktionalen Anforderungen für die zwei Automationsmodi assistiertes und teilautomatisiertes Fahren.

Wie bereits zu Beginn der Arbeit festgestellt wurde, kann keine Optimalität des Ergebnisses erreicht werden, da weder ein Verfahren zur Entwicklung eines optimalen Architekturentwurfs noch eine Bewertungsmetrik zur Validierung dieses Anspruches existiert. Dennoch zeigen die Erfahrungen in der Laufzeit des PRORETA-Projektes, insbesondere hinsichtlich der Arbeitsteilung und Wiederverwendbarkeit der entwickelten Module, dass ein Teil der Modularitätsanforderungen an das System erfüllt werden konnte. Eine Falsifikation der Referenzarchitektur im Hinblick auf die Erfüllung sowohl der funktionalen als auch nichtfunktionalen Anforderungen ist nicht erfolgt.

Die entwickelte Referenzarchitektur stellt eine dokumentierte Basis zur Weiterentwicklung von A/A-Systemen dar, in dem sie als Grundgerüst zur Implementierung solcher Systeme genutzt werden kann. Dabei ist nicht auszuschließen, dass im Zuge der Konkretisierung oder im Hinblick auf neue, noch nicht betrachtete Fahrszenarien Anpassungen an der Architektur notwendig sind. Dies steht jedoch nicht im Widerspruch zur vorliegenden Arbeit, da auch hier die Architekturentwicklung als ein iterativer Prozess gesehen wird. Die Ergebnisse der Arbeit liefern darüber hinaus einen Beitrag zur Fahrzeugsystemtechnik, in dem sie die Architektur als wissenschaftliche Disziplin in den Vordergrund stellt und dieser als Diskussionsgrundlage dient.

Weiterführende Untersuchungen sind in den Bereichen notwendig, auf die im Rahmen des PRORETA-Projekts nicht vertieft eingegangen werden konnte. Dies beinhaltet einerseits eine Umsetzung der Vollautomatisierung des Systems einschließlich einer Routenführungsfunktion. Andererseits muss sich die Referenzarchitektur in komplexen Fahrszenarien bewähren, die hohe Anforderungen an das Zusammenspiel zwischen Verhaltens- und Trajektorienplaner stellen (z. B. beim Einfädeln auf eine stark befahrene Autobahn). Darüber hinaus wird davon ausgegangen, dass die Industrialisierung automatisierter Fahrzeugführungssysteme und die damit einhergehenden Anforderungen an die Verfügbarkeit und Integrität der Systemfunktionen ein wesentlicher Treiber zukünftiger Architekturentwicklungen darstellen wird. Hierbei ist insbesondere Augenmerk auf die Beherrschung der zeitvarianten Leistungsfähigkeit des Gesamtsystems einschließlich Diagnose- und Überwachungsfunktionen zu richten, was eine Detaillierung des Konzepts des Modusplaners notwendig macht.

A Anforderungen

A.1 Szenarienkatalog für Autobahnen

Tabelle A-1: Szenarienkatalog für Autobahnen; Abkürzung: FS: Fahrstreifen; FSW: Fahrstreifenwechsel; VO: Verkehrsobjekt

Klassifikation	Nr.	Beschreibung	Verhaltensweise	Anmerkung
Knotenpunktfrei	Freifahrt	1	Fahren auf einem FS mit Wunschgeschwindigkeit	FS Befahren
		2	FSW links auf nächsten durchgehenden FS, Zusatzfahrstreifen oder neu hinzukommenden durchgehenden FS	FSW (links)
		3	FSW rechts auf nächsten durchgehenden FS	FSW (rechts)
		4	FSW rechts unter Zwang aufgrund Ende des Zusatz-FS oder aufgrund FS-Reduktion	FSW (rechts) Zwang
		5	Ausfädeln auf Ausfädelungstreifen (Rastanlage)	Ausfädeln
		6	Einfädeln auf durchgehenden FS von Einfädelungstreifen (unter Zwang)	Einfädeln Zwang
		7	FSW rechts auf Seitenstreifen	FSW (rechts)
		8	Halten auf Seitenstreifen	Halten
	Interaktion	9	Fahren auf einem FS mit Einhalten eines Sicherheitsabstandes zum Vorausfahrenden	FS Befahren
		10	Fahren auf einem FS und anderes VO von links auf eigenen FS wechseln lassen	FSW zulassen (v. links)
		11	Fahren auf einem FS und anderes VO von rechts auf eigenen FS wechseln lassen	FSW zulassen (v. rechts)
		12	Fahren auf rechtem durchgehenden FS und anderes VO von rechts auf eigenen FS einfädeln lassen	Einfädeln zulassen (v. rechts)
		13	FSW links auf nächsten durchgehenden FS, Zusatzfahrstreifen oder neu hinzukommenden durchgehenden FS in geeignete Lücke anderer VO auf Ziel-FS	FSW (links)
		14	FSW rechts auf nächsten durchgehenden FS in geeignete Lücke anderer VO auf Ziel-FS	FSW (rechts)
		15	FSW rechts unter Zwang aufgrund Ende des Zusatz-FS oder aufgrund FS-Reduktion in geeignete Lücke anderer VO auf Ziel-FS	FSW (rechts) Zwang
		16	Einfädeln auf durchgehenden FS von Einfädelungstreifen (unter Zwang) in geeignete Lücke anderer VO auf Ziel-FS	Einfädeln Zwang

Tabelle A-1: Szenarienatalog für Autobahnen (Fortsetzung); Abkürzung: FS: Fahrstreifen; FSW: Fahrstreifenwechsel; VO: Verkehrsobjekt

Klassifikation	Nr.	Beschreibung	Verhaltensweise	Anmerkung
Knotenpunkt	Freifahrt	17	Fahren auf einem FS (Verflechtungsstreifen oder Rampe)	FS Befahren
		18	Ausfädeln auf Ausfädelungsstreifen (Ausfahrt)	Ausfädeln
		19	Einfädeln auf durchgehenden FS von Einfädelungsstreifen (unter Zwang)	Einfädeln
		20	FSW rechts auf Verflechtungsstreifen	FSW (rechts)
		21	FSW links von Verflechtungsstreifen auf durchgehenden FS oder zusätzlichen FS der Verteilerfahrbahn	FSW (links)
	Interaktion	22	Fahren auf einem FS (Verflechtungsstreifen oder Rampe) mit Einhalten eines Sicherheitsabstandes zum Vorausfahrenden	FS Befahren
		23	Fahren auf einem durchgehenden FS und anderes VO von rechts auf eigenen FS einfädeln lassen	Einfädeln zulassen (v. rechts)
		24	Fahren auf einem Verflechtungsstreifen und anderes VO von rechts auf eigenen FS wechseln lassen	FSW zulassen (v. rechts)
		25	Einfädeln auf durchgehenden FS von Einfädelungsstreifen (unter Zwang) in geeignete Lücke anderer VO auf Ziel-FS	Einfädeln
		26	FSW rechts auf Verflechtungsstreifen in geeignete Lücke anderer VO auf Ziel-FS	FSW (rechts)
		27	FSW links von Verflechtungsstreifen auf durchgehenden FS oder zusätzlichen FS der Verteilerfahrbahn in geeignete Lücke anderer VO auf Ziel-FS	FSW (links)

Tabelle A-2: Szenarien katalog für Autobahnen (komprimiert); Abkürzung: FS: Fahrstreifen; VO: Verkehrsobjekt

Nr.	Verhaltensweise	Parameter	Beschreibung
1	Fahrstreifen befahren	mit / ohne unmittelbar vorausfahrendes VO	Fahren auf einem FS, Verflechtungsstreifen oder einer Rampe mit Wunschgeschwindigkeit und ggf. unter Einhaltung eines sicheren Abstandes zum vorausfahrenden VO
	Fahrstreifenwechsel zulassen (Interaktion)	von VO links, von VO rechts	Fahren auf einem FS und anderes VO von links / rechts durch aktive Anpassung der Längsgeschwindigkeit auf eigenen FS wechseln lassen (auch Einfädelung)
2	Fahrstreifenwechsel (allgemein)	links, rechts	FSW auf nächsten durchgehenden FS, Zusatzfahrstreifen, Ausfädelungsstreifen, Verflechtungsstreifen sowie Seitenstreifen
	Fahrstreifenwechsel (Interaktion)	links, rechts	FSW auf nächsten FS in geeignete Lücke anderer VO auf dem Ziel-FS
	Fahrstreifenwechsel (unter Zwang)	links, rechts	FSW rechts auf nächsten durchgehenden FS aufgrund Ende des aktuellen FS oder Zusatz-FS oder Verflechtungsstreifens bzw. FSW links auf durchgehenden FS aufgrund Ende des Einfädelungsstreifens oder Verflechtungsstreifens
3	Halten		Halten auf Seitenstreifen

A.2 Szenarienatalog für Landstraßen

Tabelle A-3: Szenarienatalog für Landstraßen; Abkürzung: FS: Fahrstreifen; FSW: Fahrstreifenwechsel; VO: Verkehrsobjekt

Klassifikation	Nr.	Beschreibung	Verhaltensweise	Anmerkung
Knotenpunktfrei	Freifahrt	1	Fahren auf einem FS mit Wunschgeschwindigkeit	FS Befahren
		2	FSW links auf Überhol-FS	FSW (links)
		3	FSW rechts auf durchgehenden FS	FSW (rechts)
		4	FSW rechts unter Zwang aufgrund Ende des Überhol-FS	FSW (rechts) Zwang
		5	Ausfädeln auf Ausfädelungstreifen (Rastanlage)	Ausfädeln
		6	Einfädeln auf durchgehenden FS von Einfädelungstreifen (unter Zwang)	Einfädeln Zwang
		7	Halten in Nothaltebucht	Halten (+ Ausfädeln)
	Interaktion	8	Fahren auf einem FS mit Einhalten eines Sicherheitsabstandes zum Vorausfahrenden	FS Befahren
		9	Fahren auf einem FS und anderes VO von links auf eigenen FS wechseln lassen	FSW zulassen (v. links)
		10	Fahren auf einem FS und anderes VO von rechts auf eigenen FS wechseln lassen	FSW zulassen (v. rechts)
		11	Fahren auf durchgehenden FS und anderes VO von rechts auf eigenen FS einfädeln lassen (Rastanlage)	Einfädeln zulassen (v. rechts)
		12	FSW links auf Überhol-FS in geeignete Lücke anderer VO auf Ziel-FS	FSW (links)
		13	FSW rechts auf durchgehenden FS in geeignete Lücke anderer VO auf Ziel-FS	FSW (rechts)
		14	FSW rechts unter Zwang aufgrund Ende des Überhol-FS in geeignete Lücke anderer VO auf Ziel-FS	FSW (rechts) Zwang
		15	Einfädeln auf durchgehenden FS von Einfädelungstreifen (unter Zwang) in geeignete Lücke anderer VO auf Ziel-FS	Einfädeln Zwang

Tabelle A-3: Szenarienkatalog für Landstraßen (Fortsetzung); Abkürzung: FS: Fahrstreifen; FSW: Fahrstreifenwechsel; LSA: Lichtsignalanlage; VO: Verkehrsobjekt

Klassifikation	Nr.	Beschreibung	Verhaltensweise	Anmerkung
Knotenpunkt planfrei	Freifahrt	16 Fahren auf einem FS (Rampe)	FS Befahren	
		17 Ausfädeln auf Ausfädelungsstreifen (Ausfahrt)	Ausfädeln	
		18 Einfädeln auf durchgehenden FS von Einfädelungsstreifen (unter Zwang)	Einfädeln	Zwang
	Interaktion	19 Fahren auf einem FS (Rampe) mit Einhalten eines Sicherheitsabstandes zum Vordermann	FS Befahren	
		20 Fahren auf einem durchgehenden FS und anderes VO von rechts auf eigenen FS einfädeln lassen	Einfädeln zulassen (v. rechts)	
		21 Einfädeln auf durchgehenden FS von Einfädelungsstreifen (unter Zwang) in geeignete Lücke anderer VO auf Ziel-FS	Einfädeln	Zwang
Knotenpunkt plangleich: Kreuzung	Freifahrt	22 Fahren auf einem FS und der Kreuzung Annähern	FS Befahren / Annähern / ggf. anhalten	
		23 FSW links auf einen 2. durchgehenden FS	FSW (links)	
		24 FSW rechts auf einen 2. durchgehenden FS	FSW (rechts)	
		25 FSW links auf einen Abbiege-FS	FSW (links)	ggf. unter Zwang wegen Richtungsentscheidung
		26 FSW rechts auf einen Abbiege-FS	FSW (rechts)	ggf. unter Zwang wegen Richtungsentscheidung
		27 In Fahrstreifen einordnen, wenn kein durchgehender FS vorhanden ist	Einordnen in FS	
		28 An Kreuzung links abbiegen unter Beachtung der Vorfahrtsregelung	Abbiegen (links)	Auch Beachtung von Radweg; Auch halb in Kreuzung einfahren und voreinander abbiegen, falls keine LSA
		29 An Kreuzung rechts abbiegen unter Beachtung der Vorfahrtsregelung	Abbiegen (rechts)	Auch Beachtung von Radweg
		30 Kreuzung kreuzen (geradeaus) unter Beachtung der Vorfahrtsregelung	Kreuzen	
		31 FSW rechts auf durchgehenden FS bei Ende des 2. durchgehenden FS (unter Zwang)	FSW (rechts)	Zwang

Tabelle A-3: Szenarienkatalog für Landstraßen (Fortsetzung); Abkürzung: FS: Fahrstreifen; FSW: Fahrstreifenwechsel; VO: Verkehrsobjekt

Klassifikation			Nr.	Beschreibung	Verhaltensweise	Anmerkung
Knotenpunkt	Plangleich: Kreuzung	Interaktion	32	Fahren auf einem FS und der Kreuzung Annähern mit Einhaltung eines Sicherheitsabstandes zum Vorausfahrenden	FS Befahren / Annähern / ggf. anhalten	
			33	FSW links auf einen 2. durchgehenden FS in geeignete Lücke anderer VO auf Ziel-FS	FSW (links)	
			34	FSW rechts auf einen 2. durchgehenden FS in geeignete Lücke anderer VO auf Ziel-FS	FSW (rechts)	
			35	FSW rechts auf durchgehenden FS bei Ende des 2. durchgehenden FS (unter Zwang) in geeignete Lücke anderer VO auf Ziel-FS	FSW (rechts)	Zwang
	plangleich: Kreisverkehr	Freifahrt	36	Fahren auf einem FS und dem Kreisverkehr Annähern	FS Befahren / Annähern / ggf. anhalten	
			37	In Kreisverkehr einfahren unter Berücksichtigung der Vorfahrtsregeln	Einfahren (Kreisverkehr)	
			38	Im Kreisverkehr auf FS fahren	FS Befahren (Kreisverkehr)	
			39	FSW (links) auf innenliegenden 2. Kreisverkehr-FS	FSW links (Kreisverkehr)	
			40	FSW (rechts) auf außenliegenden Kreisverkehr-FS	FSW rechts (Kreisverkehr)	
			41	Aus Kreisverkehr ausfahren	Ausfahren (Kreisverkehr)	
			42	Auf Bypass-FS ausfädeln	Ausfädeln	
			43	Bypass befahren	FS Befahren	
			44	Von Bypass-FS auf Haupt-FS einfädeln (unter Zwang)	Einfädeln	Zwang
		Interaktion	45	Fahren auf einem FS und dem Kreisverkehr Annähern mit Einhaltung eines Sicherheitsabstandes zum Vorausfahrenden	FS Befahren / Annähern / ggf. anhalten	
			46	Im Kreisverkehr auf innerem FS fahren und anderem VO FSW (links) ermöglichen	FSW zulassen (v. rechts)	
			47	Im Kreisverkehr auf äußerem FS fahren und anderem VO FSW (rechts) ermöglichen	FSW zulassen (v. links)	
			48	Von Bypass-FS auf Haupt-FS einfädeln (unter Zwang) in geeignete Lücke anderer VO auf Ziel-FS	Einfädeln	Zwang
			49	Fahren auf einem durchgehenden FS und anderes VO von rechts auf eigenen FS einfädeln lassen (Bypass)	Einfädeln zulassen (v. rechts)	

Tabelle A-4: Szenarien katalog für Landstraßen (komprimiert); Abkürzung: FS: Fahrstreifen; FSW: Fahrstreifenwechsel; VO: Verkehrsobjekt

Nr.	Verhaltensweise	Parameter	Beschreibung
1	Fahrstreifen befahren	mit / ohne unmittelbar vorausfahrendes VO	Fahren auf einem FS oder einer Rampe mit Wunschgeschwindigkeit und ggf. unter Einhaltung eines sicheren Abstandes zum vorausfahrenden VO. Im Kreisverkehr auf Kreisfahrbahn oder Bypass fahren
	Fahrstreifenwechsel zulassen (Interaktion)	von VO links, von VO rechts	Fahren auf einem FS und anderes VO von links / rechts durch aktive Anpassung der Längsgeschwindigkeit auf eigenen FS wechseln lassen (auch Einfädelung)
	Annähern an Knoten		Dem plangleichen Knotenpunkt annähern und Längsgeschwindigkeit den Erfordernissen der Vorfahrtsregelung anpassen (ggf. verlangsamen, anhalten)
2	In Fahrstreifen einordnen	links, rechts	An Kreuzungen in gewünschten FS einordnen, wenn keine durchgehende FS-Markierung vorhanden ist (i.d.R. bei Einbiegungen)
3	Fahrstreifenwechsel (allgemein)	links, rechts	FSW auf durchgehenden FS, Überhol-FS, Ausfädelungsstreifen oder weiteren durchgehenden FS an Kreuzungen. In zweistreifigen Kreisverkehren auf innen- bzw. außenliegenden FS wechseln oder auf Bypass wechseln (ausfädeln)
	Fahrstreifenwechsel (Interaktion)	links, rechts	FSW auf nächsten FS in geeignete Lücke anderer VO auf dem Ziel-FS
	Fahrstreifenwechsel (unter Zwang)	links, rechts	FSW rechts auf durchgehenden FS aufgrund Ende des Überhol-FS bzw. FSW links auf durchgehenden FS aufgrund Ende des Einfädelungsstreifens. Bei Kreuzungen FSW auf Abbiege-FS. An Kreisverkehren von Bypass auf durchgehenden FS wechseln (einfädeln)
4	Halten		Halten in Nothaltebucht
5	Abbiegen	links, rechts	An Kreuzungen abbiegen unter Beachtung der Vorfahrtsregelung
6	Kreuzen		Kreuzung kreuzen (geradeaus fahren) unter Beachtung der Vorfahrtsregelung
7	In Kreisverkehr ein- bzw. ausfahren	Einfahren, Ausfahren	In Kreisverkehr unter Beachtung der Vorfahrtsregelung ein- bzw. ausfahren

A.3 Szenarienatalog für Stadtstraßen

Tabelle A-5: Szenarienatalog für Stadtstraßen; Abkürzung: FS: Fahrstreifen; FSW: Fahrstreifenwechsel; VO: Verkehrsobjekt

Klassifikation	Nr.	Beschreibung	Verhaltensweise	Anmerkung
Knotenpunktfrei	Freifahrt	1	Fahren auf einem FS mit Wunschgeschwindigkeit	FS Befahren
		2	FSW links auf einen weiter links liegenden FS	FSW (links)
		3	FSW rechts auf einen weiter rechts liegenden FS	FSW (rechts)
		4	FSW rechts unter Zwang aufgrund Ende des aktuellen FS	FSW (rechts) Zwang
	Interaktion	5	Fahren auf einem FS mit Einhalten eines Sicherheitsabstandes zum Vorausfahrenden	FS Befahren
		6	Fahren auf einem FS und anderes VO von links auf eigenen FS wechseln lassen	FSW zulassen (v. links)
		7	Fahren auf einem FS und anderes VO von rechts auf eigenen FS wechseln lassen	FSW zulassen (v. rechts)
		8	FSW links auf einen weiter links liegenden FS in geeignete Lücke anderer VO auf Ziel-FS	FSW (links)
		9	FSW rechts auf einen weiter rechts liegenden FS in geeignete Lücke anderer VO auf Ziel-FS	FSW (rechts)
		10	FSW rechts unter Zwang aufgrund Ende des aktuellen FS in geeignete Lücke anderer VO auf Ziel-FS	FSW (rechts) Zwang
Knotenpunkt plangleich: Kreuzung	Freifahrt	11	Fahren auf einem FS und der Kreuzung Annähern	FS Befahren / Annähern / ggf. anhalten
		12	FSW links auf einen weiter links liegenden FS	FSW (links)
		13	FSW rechts auf einen weiter rechts liegenden FS	FSW (rechts)
		14	FSW links auf einen Abbiege-FS	FSW (links) (Zwang)
		15	FSW rechts auf einen Abbiege-FS	FSW (rechts) (Zwang)
		16	In Fahrstreifen einordnen, wenn kein durchgehender FS vorhanden ist	Einordnen in FS
		17	An Kreuzung links abbiegen unter Beachtung der Vorfahrtsregelung	Abbiegen (links) Ggf. Radweg;
		18	An Kreuzung rechts abbiegen unter Beachtung der Vorfahrtsregelung	Abbiegen (rechts) Ggf. Radweg
		19	Kreuzung kreuzen (geradeaus) unter Beachtung der Vorfahrtsregelung	Kreuzen
		20	Von Linksabbiege-FS oder Wendefahrbahn aus wenden unter Beachtung der Vorfahrtsregelung	Wenden

Tabelle A-5: Szenarienatalog für Stadtstraßen (Fortsetzung); Abkürzung: FS: Fahrstreifen; FSW: Fahrstreifenwechsel; VO: Verkehrsobjekt

Klassifikation			Nr.	Beschreibung	Verhaltensweise	Anmerkung
Knotenpunkt	plangleich: Kreuzung	Interaktion	21	Fahren auf einem FS und der Kreuzung Annähern mit Einhaltung eines Sicherheitsabstandes zum Vorfahrenden	FS Befahren / Annähern / ggf. anhalten	
			22	FSW links auf einen weiter links liegenden FS in geeignete Lücke anderer VO auf Ziel-FS	FSW (links)	
			23	FSW rechts auf einen weiter rechts liegenden FS in geeignete Lücke anderer VO auf Ziel-FS	FSW (rechts)	
			24	FSW links auf einen Abbiege-FS unter Beachtung eines Radfahrers auf Radfahrstreifen (links)	FSW (links)	(Zwang)
			25	FSW rechts auf einen Abbiege-FS unter Beachtung eines Radfahrers auf Radfahrstreifen (rechts)	FSW (rechts)	(Zwang)
	plangleich: Kreisverkehr	Freifahrt	26	Fahren auf einem FS und dem Kreisverkehr Annähern	FS Befahren / Annähern / ggf. anhalten	
			27	In Kreisverkehr einfahren unter Berücksichtigung der Vorfahrtsregeln	Einfahren (Kreisverkehr)	
			28	Im Kreisverkehr auf FS fahren	FS Befahren (Kreisverkehr)	
			29	FSW (links) auf innenliegenden 2. Kreisverkehr-FS	FSW links (Kreisverkehr)	
			30	FSW (rechts) auf außenliegenden Kreisverkehr-FS	FSW rechts (Kreisverkehr)	
			31	Aus Kreisverkehr ausfahren	Ausfahren (Kreisverkehr)	
			32	Auf Bypass-FS ausfädeln	Ausfädeln	
			33	Bypass befahren	FS Befahren	
			34	Von Bypass-FS auf Haupt-FS einfädeln (unter Zwang)	Einfädeln	Zwang
		Interaktion	35	Fahren auf einem FS und dem Kreisverkehr Annähern mit Einhaltung eines Sicherheitsabstandes zum Vorfahrenden	FS Befahren / Annähern / ggf. anhalten	
			36	Im Kreisverkehr auf innerem FS fahren und anderem VO FSW (links) ermöglichen	FSW zulassen (v. rechts)	
			37	Im Kreisverkehr auf äußerem FS fahren und anderem VO FSW (rechts) ermöglichen	FSW zulassen (v. links)	
			38	Von Bypass-FS auf Haupt-FS einfädeln (unter Zwang) in geeignete Lücke anderer VO auf Ziel-FS	Einfädeln	Zwang
			39	Fahren auf einem durchgehenden FS und anderes VO von rechts auf eigenen FS einfädeln lassen (Bypass)	Einfädeln zulassen (v. rechts)	
		Sondersituationen			40	An Fußgängerüberweg annähern und ggf. anhalten

Tabelle A-6: Szenarienkatalog für Stadtstraßen (komprimiert); Abkürzung: FS: Fahrstreifen; FSW: Fahrstreifenwechsel; VO: Verkehrsobjekt

Nr.	Verhaltensweise	Parameter	Beschreibung
1	Fahrstreifen befahren	mit / ohne unmittelbar vorausfahrendes VO	Fahren auf einem FS mit Wunschgeschwindigkeit und ggf. unter Einhaltung eines sicheren Abstandes zum vorausfahrenden VO. Im Kreisverkehr auf Kreisfahrbahn oder Bypass fahren
	Fahrstreifenwechsel zulassen	von VO links, von VO rechts	Fahren auf einem FS und anderes VO von links / rechts durch aktive Anpassung der Längsgeschwindigkeit auf eigenen FS wechseln lassen (auch Einfädelung)
	Annähern an Knoten / Fußgängerüberweg		Dem plangleichen Knotenpunkt annähern und Längsgeschwindigkeit den Erfordernissen der Vorfahrtsregelung anpassen (ggf. verlangsamen, anhalten)
2	In Fahrstreifen einordnen	links, rechts	An Kreuzungen in gewünschten FS einordnen, wenn keine durchgehende FS-Markierung vorhanden ist (i.d.R. bei Einbiegungen)
3	Fahrstreifenwechsel (allgemein)	links, rechts	FSW auf durchgehenden FS (auch an Kreuzungen). In zweistreifigen Kreisverkehren auf innen- bzw. außenliegenden FS wechseln oder auf Bypass wechseln (ausfädeln)
	Fahrstreifenwechsel (Interaktion)	links, rechts	FSW auf nächsten FS in geeignete Lücke anderer VO auf dem Ziel-FS. An Kreuzungen ggf. unter Beachtung von Radfahrstreifen
	Fahrstreifenwechsel (unter Zwang)	links, rechts	FSW rechts auf durchgehenden FS aufgrund Ende des aktuellen FS. Bei Kreuzungen FSW auf Abbiege-FS. An Kreisverkehren von Bypass auf durchgehenden FS wechseln (einfädeln)
4	Abbiegen	links, rechts	An Kreuzungen abbiegen unter Beachtung der Vorfahrtsregelung
5	Kreuzen		Kreuzung kreuzen (geradeaus fahren) unter Beachtung der Vorfahrtsregelung
6	Wenden		Von Linksabbiege-FS oder Wendefahrbahn aus unter der Beachtung der Vorfahrtstregelung wenden
7	In Kreisverkehr ein- bzw. ausfahren	Einfahren, Ausfahren	In Kreisverkehr unter Beachtung der Vorfahrtsregelung ein- bzw. ausfahren

A.4 Szenarienatalog für assistiertes und unfallvermeidendes Fahren

Tabelle A-7: Szenarienatalog³⁸² für assistiertes und unfallvermeidendes Fahren; Abkürzung: FS: Fahrstreifen; VO: Verkehrsobjekt

Rang	Unfalltyp	Typennummer	Anteil	Handlungsfehler
1	Unfall im Längsverkehr	601, 602, 611	12,4%	Auffahren auf Vorfahrenden
2	Fahrerunfall	101, 102	10,6%	Verlust der Kontrolle
3	Einbiegen-/Kreuzen-Unfall	301, 302, 303	9,2%	Kollision im Kreuzungsbereich mit einem Bevorrechtigtem Fahrzeug von links
4	Fahrerunfall	141, 142, 143	8,4%	Verlust der Kontrolle
5	Einbiegen-/Kreuzen-Unfall	321, 322, 323	8,0%	Kollision im Kreuzungsbereich mit einem Bevorrechtigtem Fahrzeug von rechts
6	Einbiegen-/Kreuzen-Unfall	341, 342	6,0%	Kollision im Kreuzungsbereich mit einem Bevorrechtigtem Fahrzeug
7	Abbiege-Unfall	211	4,9%	Kollision mit Entgegenkommendem
8	Abbiege-Unfall	201, 231	4,6%	Auffahren UND Seitliche Kollision beim Überholen
9	Unfall im Längsverkehr	621, 623	3,9%	Auffahren auf Wartepflichtigen (Lichtsignalanlage)
10	Unfall im Längsverkehr	631, 641, 639, 649	3,3%	Seitliche Kollision durch FS-Wechsel links / rechts (Schuldiger: FS-Wechsler)
11	Fahrerunfall	124	1,4%	Verlust der Kontrolle
12	Abbiege-Unfall	243	1,3%	Seitliche Kollision mit Fahrradfahrer rechts
13	Fahrerunfall	121, 122	1,2%	Verlust der Kontrolle
14	Sonstiger Unfall	721	1,2%	Wenden
15	Unfall mit ruhendem Verkehr	501	0,9%	Kollision mit ruhendem Verkehr rechts
16	Abbiege-Unfall	202, 232	0,8%	Seitliche Kollision beim Abbiegen (Schuldiger: Abbieger)
17	Abbiege-Unfall	244	0,5%	Kollision mit Entgegenkommendem (Fahrradfahrer)
18	Sonstiger Unfall	723	0,3%	Wenden
19	Unfall im Längsverkehr	681	0,3%	Seitliche Kollision entgegenkommender Fahrzeuge
20	Unfall mit ruhendem Verkehr	541	0,3%	Auffahren auf ruhenden Verkehr
21	Einbiegen-/Kreuzen-Unfall	305	0,2%	Kollision im Kreuzungsbereich mit einem Bevorrechtigtem Fahrzeug
22	Unfall im Längsverkehr	651	0,2%	Seitliche Kollision (Nebeneinanderfahrende Fzge.)
23	Abbiege-Unfall	251, 252	0,1%	Seitliche Kollision gemeinsam abbiegender Fahrzeuge

³⁸² Unfalltypennummer gemäß <https://udv.de/de/initiativen-aktionen/unka>, letzter Zugriff am 09.10.16

B Schnittstellenbeschreibung

Im Folgenden werden die Schnittstellen der Module auf Basis der Bausteinsicht in Abbildung 6-1 genauer beschrieben. Dabei sei angemerkt, dass diese möglichst implementierungsunabhängig formuliert sind. Ebenfalls werden nicht alle Möglichkeiten gleichbedeutender oder ineinander überführbarer Schnittstellen aufgeführt.

Schnittstellen des Missionsplaners

Vom Weltmodell: Straßennetz

- Topologische Karte des Straßennetzes einschließlich entsprechender geometrischer und semantischer Informationsattributen (u.a. hinsichtlich Verkehrsregeln)
- Kartenrelative Ego-Pose
- Optional: Externe Daten, u.a. Verkehrsfluss oder Events, die die Routenwahl beeinflussen können

Von MMS: Ziel

- Ziel-Adresse / -Koordinate
- Optional: Routenparameter (Anm.: Analog zu herkömmlichen Navigationssystemen)
- Rückmeldung zur Routen-Präferenzabfrage (vgl. Kap. 6.2)

Zur MMS: Route

- Berechnete Route
- Routeninformationen (u.a. Distanz, geschätzte Fahrtzeit, ggf. von Automation unterstützte Teilbereiche der Route)
- Routen-Präferenzabfrage (vgl. Kap. 6.2)

An Verhaltensplaner: Ziel-Fahrstreifen

- Referenz auf Zielfahrstreifen vor und ggf. (falls nicht eindeutig) nach Knotenpunkten der Route (bspw. auf Identitätskennung der Fahrstreifenelemente der fahrstreifengenauen Karte)

Vom Verhaltensplaner: FS-Eingrenzung (s.u.)

Vom Modusplaner: Missionsvorgabe (s.u.)

An Modusplaner: Missionsstatus

-
- Routenrelative Informationen semantischer Kartenattribute (Anm.: Analog eines „elektronischen Horizonts“), die Rückschlüsse auf die Verkehrsdomäne und vom System unterstützte Szenerie-Elemente zulassen (u.a. Straßentyp)

Schnittstellen des Verhaltensplaners

Vom Weltmodell: Verkehrsszene (fahrstreifengenau)

- Fahrstreifengenaue Weltmodellierung bezogen auf lokale Ego-Pose, einschl.:
 - Semantische Zusammenhänge der Fahrstreifenelemente an Knotenpunkten
 - Geometrischer Verlauf der Fahrstreifen
 - Semantische Informationsattribute, zugehörig zu Straßenverläufen oder Fahrstreifen, die Verkehrsregel-relevant sind
 - Optional: Fahrstreifengenaue Informationsattribute aus externen Daten einschl. Events (z. B. Stau, Unfall..)
- Dynamisches Umfeld bezogen auf lokale Ego-Pose
 - Schätzung der Zustände anderer Verkehrsobjekte (Klassifizierung, Bewegungszustand)
 - Schätzung der Bewegungsprädiktion anderer Verkehrsobjekte auf Zustandsebene (einschl. der Berücksichtigung einer Manöverprädiktion)
- Lokale Ego-Pose (einschließlich daraus ableitbarer Eigenbewegungszustände)

Von MMS: Soll-Manöver

- Bezeichnung des aktuell vom Fahrer gewünschten (expliziten) Fahrmanövers

An MMS: Ist-Manöver

- Bezeichnung des aktuell vom System ausgeführten Fahrmanövers
- Optional: Manöver-Ausführungsstatus (z. B. Manöverfortschritt)
- Benennung der vom System aktuell ausführbarer Manöver (vgl. Kap. 6.2)
- Im Falle assistierten Fahrens: Verhaltensempfehlung für Fahrer (vgl. Kap. 6.2)

Von Missionssplaner: Ziel-Fahrstreifen (s.o.)

An Missionssplaner: FS-Eingrenzung

- Referenz auf ggf. nicht erreichbaren Ziel-Fahrstreifen

An Trajektorienplaner: Soll-Verhalten (Manöver)

- Soll-Verhalten...
 - entweder in Form einer expliziten Beschreibung des Verhaltens (Manöverbezeichnung, Zeitdauer, erwarteter System-Endzustand)
 - oder implizit in Form erwarteter Aufenthaltsbereiche („Checkpoints“, Zeitdauer zur Erreichung, erwarteter Systemzustand bei Erreichung)

- Verhaltensspielraum als generalisierte Angabe der max. zulässigen Kosten, die im Rahmen der Ausführung nicht überschritten werden dürfen
- *Anmerkung: Ebenfalls kann das räumliche Bezugssystem zum Verhaltensspielraum variieren (z. B. können die Werte zum erwarteten Aufenthaltsbereich absolut oder auch relativ zu anderen Verkehrsobjekten angegeben werden, was funktions- und situationsspezifisch vorteilhaft sein kann).*

Von Trajektorienplaner: Ist-Verhalten (Status) (s.u.)

Vom Modusplaner: Verhaltensvorgabe (s.u.)

An Modusplaner: Verhaltensstatus

- Bezeichnung des aktuell vom System ausgeführten Fahrmanövers
- Manöver-Ausführungsstatus (z. B. Manöverfortschritt)
- Ausgabe von Ausnahmefällen („Exceptions“),
 - Angabe, dass zur Verfügung gestellter Entscheidungsspielraum nicht ausreicht, um ein valides Manöver zu finden (s.u.: Schnittstelle Verhaltensvorgabe; s.o. Ziel-Fahrstreifen)
 - Angabe, ob ein unerwartetes, reaktives Verhalten aktiv ist (z. B. Kollisionsvermeidung)

Schnittstellen des Trajektorienplaners

Vom Weltmodell: Verkehrsszene (geometrisch)

- Quasikontinuierliche und geometrische Weltmodellierung
 - Anordnung und (geometrischer) Verlauf der Fahrstreifen
- Dynamisches Umfeld bezogen auf lokale Ego-Pose
 - Schätzung der Zustände anderer Verkehrsobjekte (Bewegungszustand)
 - Schätzung der Bewegungsprädiktion anderer Verkehrsobjekte auf Zustandsebene
- Lokale Ego-Pose (einschließlich daraus ableitbarer Eigenbewegungszustände)

Zur MMS: Trajektorie

- Optional im Falle assistierten Fahrens: Sichere Fahrtrajektorie und Sicherheitsspielraum (s. Kap. 6.3)

Vom Verhaltensplaner: Soll-Verhalten (Manöver) (s.o.)

An Verhaltensplaner: Ist-Verhalten (Status)

- Bezeichnung des aktuellen Verhaltens (Manöver)
- Ausführungsstatus (Fortschrittsmaß) des aktuellen Verhaltens (Manöver)

An Trajektorienregler: Trajektorie

- Raumzeitliche Soll-Trajektorie

-
- *Anmerkung: Trajektorie kann diskret (als Punktemenge) oder kontinuierlich (z. B. als math. beschreibbare Funktion) weitergegeben werden. Wichtig ist eine Kopplung von räumlichen und zeitlichen Verläufen.*

Schnittstellen der Stabilisierungsebene

Vom Weltmodell: Eigenzustand

- Lokale Ego-Pose (einschließlich daraus ableitbarer Eigenbewegungszustände)

Vom Trajektorienplaner: Trajektorie (s.o.)

Innerhalb: An Aktoren: Soll-Stellgrößen

- *Anmerkung: Wird nicht weiter vertieft, da Hardware-abhängig*

Schnittstellen des Modusplaners

Vom Weltmodell: Leistungsfähigkeit

- Leistungsfähigkeit der Hardware-Komponenten (Status / Diagnose von Sensoren, Aktoren, Spannungsversorgung etc.)
- Leistungsfähigkeit der Software-Komponenten (z. B. Laufzeiten)
- Kennwerte des Umfeldmodells:
 - Reichweiten
 - Unsicherheit (räumliche, Existenz-Unsicherheiten)
- Kennwerte der Betriebsmittel
- Optional: Leistungsfähigkeit des Fahrers / Operators

Von Missionssplaner: Missionsstatus (s.o.)

An Missionssplaner: Missionsvorgabe

- Aufforderung zur Initiierung einer Fahrmission (Route)

Von Verhaltensplaner: Verhaltensstatus (s.o.)

An Verhaltensplaner: Verhaltensvorgabe

- Angabe zulässiger Fahrmanöver
Angabe zulässiger Verhaltensspielräume (u.a. zulässige Geschwindigkeitsbereiche, zugrundeliegende max. Horizontalbeschleunigung)

Literaturverzeichnis

- Abowd, G.; Bass, L.; Kazman, R.; Webb, M.:** SAAM: A Method for Analyzing the Properties of Software Architectures. In: Software Engineering Institute White Paper, Carnegie Mellon University, Pittsburgh, 2007
- Albus, J.; Lumia, R.; Fiala, J.:** NASREM - The NASA/NBS Standard Reference Model for Telerobot Control System Architecture, National Institute of Standards and Technology, Gaithersburg, 1994
- Albus, J.:** The NIST Real Time Control System (RCS): An Approach to Intelligent Systems Research. In: Journal of Experimental and Theoretical Artificial Intelligence, Volume 9, Issue 2-3, 1997
- Albus, J.:** 4-D/RCS Reference Model Architecture for Unmanned Ground Vehicles. In: Proceedings of the AeroSense Session on Unmanned Ground Vehicle Technology, 2000
- Antsaklis, P. J.; Passino, K. M.:** Introduction to Intelligent and Autonomous Control with High Degree of Autonomy. In: Introduction to Intelligent and Autonomous Control, Kluwer Academic Publishers, 1992
- Arkin, R. C.:** Behavior-Based Robotics, 3. Auflage, The MIT Press, Cambridge, 2000
- Bahram, M.; Aeberhard, M.; Wollher, D.:** Please Take Over! An Analysis and Strategy For a Driver Take Over Request During Autonomous Driving. In: Proceedings of the IEEE Intelligent Vehicles Symposium, Soul, 2015
- Baker, C. R.; Ferguson D. I.; Dolan, M. J.:** Robust Mission Execution for Autonomous Urban Driving, Research Showcase, Robotics Institute, Carnegie Mellon University, Pittsburgh, 2008
- Baratoff G.; Ahrns I.; Toepfer C.; Neumann H.:** Ortsvariantes aktives Sehen: Von biologischer Motivation zu technischer Realisierung. In: Künstliche Intelligenz, 1/1999, arenDTaP Verlag, Bremen, 1999
- Bartels, A.; Rohlf, M.; Hamel, S.; Saust, F.; Klauske, L. C.:** Querführungsassistentz. In: In: Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015
- Bass, L.; Clements, P.; Kazman, R.:** Software Architecture in Practice, Addison-Wesley, Boston, 2003
- Bass, L.; Clements, P.; Kazman, R.:** Software Architecture in Practice, 3. Auflage, Pearson Education, Upper Saddle River, 2013

-
- Bauer, E.; Konigorski, U.:** Ein modellprädiktiver Querplanungsansatz zur Kollisionsvermeidung. 6. VDI Fachtagung AUTOREG, Baden-Baden, Juni 2013
- Bayouth, M.; Nourbakhsh, I.; Thorpe, C.:** A Hybrid Human-Computer Autonomous Vehicle Architecture. In: Proceedings of the 3rd ECPD International Conference on Advanced Robotics, Intelligent Automation and Control, Belgrad, 1997
- Behere, S.; Törngren, M.:** A Functional Architecture for Autonomous Driving. In: Proceedings of the First International Workshop on Automotive Software Architecture, Montreal, 2015
- Behrens, R.; Kleine-Besten, T.; Pöchmüller, W.; Engelsberg, A.:** Digitale Karten im Navigation Data Standard Format. In: Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015
- Behringer, R.; Müller, N.:** Autonomous Road Vehicle Guidance from Autobahnen to Narrow Curves. In: IEEE Transactions on Robotics and Automation, Volume 14, No. 5, 1998
- Bender, P.; Stiller, C.:** Trajektorienplanung: Manöveridentifikation anhand der Topologie des Freiraums. In: Tagungsband des 10. Workshop Fahrerassistenzsysteme, Walting, 2014
- Broggi, A.; Debattisi, S.; Panciroli, M.; Porta, P. P.:** Moving from Analog to Digital Driving. In: Proceedings of the IEEE Intelligent Vehicles Symposium IV 2013, Gold Coast, 2013
- Brooks, R.:** A Robust Layered Control System for a Mobile Robot. In: IEEE Journal of Robotics and Automation, Volume 2, No. 1, 1986
- Bubb, H.; Bengler, K.; Grünen, R. E.; Vollrath, M.:** Automobilergonomie, Springer Vieweg, Wiesbaden, 2015
- Bundesministerium für Verkehr:** Straßenverkehrsordnung (StVO), 03/2013, 2013
- Burgard, W.; Hebert, M.:** World Modeling. In: Siciliano, B.; Khatib, O.: Springer Handbook of Robotics, Springer-Verlag, Berlin, 2008
- Czerwionka, P.; Wang, M.:** Optimized Route Network Graph as Map Reference for Autonomous Cars Operating on German Autobahn, Institute of Computer Science, Freie Universität Berlin, 2012
- Darms, M.:** Eine Basis-Systemarchitektur zur Sensordatenfusion von Umfeldsensoren für Fahrerassistenzsysteme, Dissertation, Fortschritt Berichte VDI, Reihe 12 Nr. 653, VDI Verlag, Düsseldorf, 2007
- Dickmanns, E. D.; Behringer, R.; Dickmanns, D.; Hildebrandt, T.; Maurer, M.; Thomanek, F.; Schiehlen, J.:** The seeing passenger car 'VaMoRs-P'. In: Proceedings of the 1994 IEEE Intelligent Vehicles Symposium, 1994

- Dickmanns, E. D.:** Dynamic Vision for Perception and Control of Motion, Springer-Verlag, London, 2007
- Domsch, C.; Negele, H.:** Einsatz von Referenzfahrsituationen bei der Entwicklung von Fahrerassistenzsystemen. In: Tagung Aktive Sicherheit, München, 2008
- Donges, E.:** Fahrerverhaltensmodelle. In: Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015
- Düring, M.; Pascheka, P.:** Cooperative Decentralized Decision Making for Conflict Resolution among Autonomous Agents. In: Proceedings of the 2014 IEEE International Symposium on Innovations in Intelligent Systems and Applications, 2014
- Eicker, S.; Hegmanns, C.; Malich, S.:** Auswahl von Bewertungsmethoden für Softwarearchitekturen. In: ICB Research Report, Nr. 14, 2007, Institut für Informatik und Wirtschaftsinformatik der Universität Duisburg Essen, Essen, 2007
- Europäische Union:** Verordnung (EG) Nr. 661 / 2009 des Europäischen Parlamentes und des Rates vom 13. Juli 2009, Amtsblatt der Europäischen Union, 2009
- Fastenmeier, W.:** Autofahrer und Verkehrssituation, Verlag TÜV Rheinland, Köln, 1995
- Forschungsgesellschaft für Straßen- und Verkehrswesen:** Handbuch für die Bemessung von Straßenverkehrsanlagen (HBS), FGSV-Verlag, Köln, 2009
- Forschungsgesellschaft für Straßen- und Verkehrswesen:** Richtlinie für die Anlage von Autobahnen, FGSV-Verlag, Köln, 2008
- Forschungsgesellschaft für Straßen- und Verkehrswesen:** Richtlinie für die Anlage von Landstraßen, FGSV-Verlag, Köln, 2012
- Forschungsgesellschaft für Straßen- und Verkehrswesen:** Richtlinie für die Anlage von Stadtstraßen, FGSV-Verlag, Köln, 2006
- Franz, B.:** Entwicklung und Evaluation eines Interaktionskonzepts zur manöverbasierten Führung von Fahrzeugen, Dissertation, Institut für Arbeitswissenschaft der TU Darmstadt, 2014
- Franz, B.; Kauer, M.; Geyer S.; Hakuli, S.:** Conduct-by-Wire. In: Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015
- Gasser, T.; Arzt, C.; Ayoubi, M.; Bartels, A.; Eier, J.; Flemisch, F.; Häcker, D.; Hesse, T.; Huber, W.; Lotz, C.; Maurer, M.; Ruth-Schumacher, S.; Schwarz, J.; Vogt, W.:** Rechtsfolgen zunehmender Fahrzeugautomatisierung, BASt-Bericht F 83, Bundesanstalt für Straßenwesen, Bergisch-Gladbach, 2012

-
- Gasser, T.; Seeck, A.; Smith, B. W.:** Rahmenbedingungen für die Fahrerassistenzentwicklung. In: Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015
- Geyer, S.:** Entwicklung und Evaluierung eines kooperativen Interaktionskonzepts an Entscheidungspunkten für die teilautomatisierte, manöverbasierte Fahrzeugführung, Dissertation, Fortschritt Berichte VDI, Reihe 12 Nr. 770, VDI Verlag, Düsseldorf, 2013
- Geyer, S.:** Maneuver-Based Vehicle Guidance Based on the Conduct-by-Wire Principle. In: Maurer, M.; Winner, H.: Automotive Systems Engineering, Springer-Verlag, Berlin, 2013
- Geyer, S.; Baltzer, M.; Franz, B.; Hakuli, S.; Kauer, M.; Kienle, M.; Meier, S.; Weißberger, T.; Bengler, K.; Bruder, R.; Flemisch, F. O.; Winner, H.:** Concept and Development of a Unified Ontology for Generating Test and Use Case Catalogues for Assisted and Automated Vehicle Guidance. In: IET Intelligent Transport Systems, Volume 8, Issue 3, 2014
- Graupner, M.:** Bewertung des Standes der Forschung zum Kooperationsverhalten im Straßenverkehr und Ableitung von Anforderungen an das Kooperationsverhalten bei Nutzung höher automatisierter Fahrzeuge, Master-Thesis, Fachgebiet Fahrzeugtechnik der TU Darmstadt, 2013
- Grewe, R.:** Optimierung der Repräsentation von Occupancy-Grids für Fahrerassistenzsysteme, Dissertation, Fortschritt Berichte VDI, Reihe 12 Nr. 780, VDI Verlag, Düsseldorf, 2014
- Grünig, R.; Kühn, R.:** Entscheidungsverfahren für komplexe Probleme, 4. Auflage, Springer-Verlag, Berlin, 2013
- Hakuli, S.; Geyer, S.; Winner, H.; Kauer, M.; Schreiber, M.:** Systematic Development of Manoeuvre-Based Driver Assistance Functions for Conduct-by-Wire with IPG CarMaker. In: Apply & Innovate 2010, Karlsruhe, 2010
- Hakuli, S.; Geyer, S.; Winner, H.; Henning, J.:** Integriertes Konzept für die manöverbasierte Fahrerassistenz. In: ATZ Automobiltechnische Zeitschrift, Ausgabe 3-2011, 2011
- Hakuli, S.; Krug, M.:** Virtuelle Integration. In: Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015
- Hamdi, M. S.:** Entwurf adaptiver lernender Roboter, Dissertation, Fachbereich Informatik der Universität Hamburg, 1999

- Hawranek, D.; Schulz, T.:** Die zweite Erfindung des Automobils. In: Der SPIEGEL Wissen, Nr. 4/2014: Das Auto von Morgen, SPIEGEL-Verlag, Hamburg, 2014
- Hertzberg, J.; Chatila, R.:** AI Reasoning Methods for Robotics. In: Siciliano, B.; Khatib, O.: Springer Handbook of Robotics, Springer-Verlag, Berlin, 2008
- Hertzberg, J.; Lingemann, K.; Nüchter, A.:** Mobile Roboter, Springer-Verlag, Berlin, 2012
- Hillenbrand, M.:** Funktionale Sicherheit nach ISO 26262 in der Konzeptphase der Entwicklung von Elektrik/Elektronik Architekturen von Fahrzeugen, KIT Scientific Publishing, Karlsruhe, 2012
- Hohm, A.; Lotz, F.; Fochler, O.; Lücke, S.; Winner, H.:** Automated Driving in Real Traffic: From Current Technical Approaches towards Architectural Perspectives. In: SAE Technical Paper 2014-01-0159, SAE World Congress 2014, Detroit, 2014
- Ibanez-Guzman, J.; Laugier, C.; Yoder, J.-D., Thrun, S.:** Autonomous Driving: Context and State-of-the-Art. In: Eskandarian, Azim: Handbook of Intelligent Vehicles, Springer, London, 2012
- Institute of Electrical and Electronics Engineers, Circuit Theory Group:** IEEE Standard 1471-2000 - Recommended practice for architectural description of software-intensive systems, 2000
- International Organization for Standardization:** ISO 26262: Road Vehicles - Functional Safety, 2011
- International Organization for Standardization:** ISO 14825: Intelligent Transport Systems - Geographic Data Files (GDF), 2011
- International Organization for Standardization:** ISO/IEC 9126: Software Engineering - Product Quality, 2012
- International Organization for Standardization:** ISO/IEC 25000: Systems- and Software Engineering, 2014
- Kauer, M.; Schreiber, M.; Bruder, R.:** How to conduct a car? In: Proceedings of the IEEE Intelligent Vehicles Symposium, Baden-Baden, 2011
- Kazman, R.; Klein, M.; Clements, P.:** ATAM: Method for Architecture Evaluation. In: Technical Report, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 2000
- Kim, S.; Kim, K.; Kim, T.-H.:** Human-Aided Cleaning Algorithm for Low-Cost Robot Architecture. In: Human-Computer Interaction, Part III, Springer-Verlag, Berlin, 2007

-
- Klanner, F.; Ruhhammer, C.:** Backendsysteme zur Erweiterung der Wahrnehmungsreichweite von Fahrerassistenzsystemen. In: Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015
- Kleine-Besten, T.; Kersken, U.; Pöchmüller, W.; Schepers, H.; Mlasko, T.; Behrens, R.; Engelsberg, A.:** Navigation und Verkehrstelematik. In: Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015
- Kohlhaas, R.; Bittner, T.; Schamm, T.; Zöllner, M.:** Semantic State Space for High-Level Maneuver Planning in Structured Traffic Scenes. In: Proceedings of the 2014 International Conference on Intelligent Transportation Systems Conference, Qingdao, 2014
- Kortenkamp, D.; Simmons, R.:** Robotic Systems Architectures and Programming. In: Siciliano, B.; Khatib, O.: Springer Handbook of Robotics, Springer-Verlag, Berlin, 2008
- Langer, I.; Abendroth, B.; Bruder, R.:** Fahrerzustandserkennung. In: Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015
- Lotz, F.:** System Architectures for Automated Vehicle Guidance Concepts. In: Maurer, M.; Winner, H.: Automotive Systems Engineering, Springer-Verlag, Berlin, 2013
- Lotz, F.; Winner, H.:** Maneuver Delegation and Planning for Automated Vehicles at Multi-Lane Road Intersections. In: Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems, Qingdao, 2014
- Löper, C.; Kelsch, J.; Flemisch, F.:** Kooperative, manöverbasierte Automation und Arbitrierung als Bausteine für hochautomatisiertes Fahren. In: Gesamtzentrum für Verkehr Braunschweig: Automatisierungs-, Assistenzsysteme und eingebettete Systeme für Transportmittel, 2008
- Matthaei, R.; Reschka, A.; Rieken, J.; Dierkes, F.; Ulbrich, S.; Winkle, T.; Maurer, M.:** Autonomes Fahren. In: Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015
- Matthaei, R.:** Wahrnehmungsgestützte Lokalisierung in fahrstreifengenauen Karten, Dissertation, TU Braunschweig, Shaker Verlag, Aachen, 2015
- Maurer, M.:** Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen, Dissertation, Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München, 2000

- Maurer, M.:** Entwurf und Test von Fahrerassistenzsystemen. In: Winner, H.; Hakuli, S.; Wolf, G.: Handbuch Fahrerassistenzsysteme, 2. Auflage, Vieweg+Teubner Verlag, Wiesbaden, 2012
- Maurer, M.:** Automotive Systems Engineering: A Personal Perspective. In: Maurer, M.; Winner, H.: Automotive Systems Engineering, Springer-Verlag, Berlin, 2013
- Mesarovic, M.; Macko, D.; Takahara, Y.:** Theory of Hierarchical, Multilevel Systems. Academic Press, New York, 1970
- Nothdurft, T.; Hecker, P.; Ohl, S.; Saust, F.; Maurer, M.; Reschka, A.; Böhmer, J. R.:** Stadtpilot: First Fully Autonomous Test Drives in Urban Traffic. In: IEEE, Proceedings of the IEEE Conference on Intelligent Transportation Systems (ITSC), Washington, 2011
- Omasreiter, H.; Metzker, E.:** A Context-Driven Use Case Creation Process for Specifying Automotive Driver Assistance Systems. In: Proceedings of the 12th IEEE International Requirements Engineering Conference, Kyoto, 2004
- Pellkofer, M.:** Verhaltensentscheidung für autonome Fahrzeuge mit Blickrichtungssteuerung, Dissertation, Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München, 2003
- Pfromm, M.; Cieler, S.; Bruder, R.:** Driver Assistance via Optical Information with Spatial Reference. In: Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems, Den Haag, 2013
- Pfromm, M.:** Menschbezogener Umgang mit Systemstörungen bei teilautomatisierter Fahrzeugführung, Dissertation, Institut für Arbeitswissenschaft der TU Darmstadt, in Druck, 2016
- Posch, T.; Birken, K.; Gerdorf, M.:** Basiswissen Softwarearchitektur, dpunkt-Verlag, Heidelberg, 2007
- Rasmussen, J.:** Skills, Rules and Knowledge; Signals, Signs and Symbols, and Other Distinctions in Human Performance Models. In: IEEE Transactions on Systems, Man and Cybernetics, 1983
- Rauskolb, F.; Berger, K.; Lipski, C.; Magnor, M.; Cornelsen, K.; Effertz, J.; Form, T.; Graefe, F.; Ohl, S.; Schumacher, W.; Wille, J.-M.; Hecker, P.; Nothdurft, T.; Doering, M.; Homeier, K.; Morgenroth, J.; Wolf, L.; Basarke, C.; Berger, C.; Gülke, T.; Klose, K.; Rumpe, B.:** Caroline: An Autonomously Driving Vehicle for Urban Environments. In: Journal of Field Robotics, Volume 25, Issue 9, 2008
- Reichardt, D.:** Kontinuierliche Verhaltenssteuerung eines autonomen Fahrzeugs in dynamischer Umgebung, Dissertation, Fachbereich Informatik der Universität Kaiserslautern, 1996

-
- Reichart & Bielefeld (2012):** Einflüsse von Fahrerassistenzsystemen auf die Systemarchitektur im Kraftfahrzeug. In: Winner, H.; Hakuli, S.; Wolf, G.: Handbuch Fahrerassistenzsysteme, 2. Auflage, Vieweg+Teubner Verlag, Wiesbaden, 2012
- Reinards, M.; Kormann, G.; Scheff, U.:** Fahrerassistenzsysteme bei Traktoren. In: Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015
- Reschka, A.; Rieken, J.; Maurer, M.:** Entwicklungsprozess von Kollisionsschutzsystemen. In: Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015
- Rosenblatt, J.:** DAMN: A Distributed Architecture for Mobile Navigation. In: Journal of Experimental & Theoretical Artificial Intelligence, Volume 9, Issue 2-3, 1997
- Society of Automotive Engineers:** SAE Standard J3016: Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems, SAE International, 2014
- Saust, F.; Müller, T.; Wille, J. M.; Maurer, M.:** Entwicklungsbegleitendes Simulations- und Testkonzept für autonome Fahrzeuge in städtischen Umgebungen, Institut für Regelungstechnik der TU Braunschweig, Braunschweig, 2009
- Schäuffele, J.; Zurawka, T.:** Automotive Software Engineering, 5. Auflage, Springer Vieweg, Wiesbaden, 2012
- Schönichen, J.:** Integration von Informationen digitaler Karten in ein Fusions-Framework zur Verbesserung der Umfeldwahrnehmung für Fahrerassistenz-Konzepte, Master-Thesis, Fachgebiet Regelungstheorie und Robotik der TU Darmstadt, 2012
- Schreiber, M.:** Konzeptionierung und Evaluierung eines Ansatzes zu einer manöverbasierten Fahrzeugführung im Nutzungskontext Autobahnfahrten, Dissertation, Institut für Arbeitswissenschaft der TU Darmstadt, 2011
- Schreier, M.:** Bayesian Environment Representation, Prediction, and Criticality Assessment for Driver Assistance Systems, Dissertation, Fortschritt Berichte VDI, Reihe 12 Nr. 797, VDI Verlag, Düsseldorf, 2016
- Schuldt, F.; Lichte, B.; Maurer, M.; Scholz, S.:** Systematische Auswertung von Testfällen für Fahrfunktionen im modularen virtuellen Testbaukasten. In: Tagungsband des 9. Workshop Fahrerassistenzsysteme, Walting, 2014
- Schwarting, W.; Pascheka, P.:** Recursive Conflict Resolution for Cooperative Motion Planning in Dynamic Highway Traffic. In: Proceedings of the 2014 IEEE Intelligent Transportation Systems, Qingdao, 2014

- Siedersberger, K.-H.:** Komponenten zur automatischen Fahrzeugführung in sehenden (semi-)autonomen Fahrzeugen, Dissertation, Universität der Bundeswehr, München, 2003
- Starke, G.:** Effektive Software-Architekturen, 4. Auflage, Hanser, München, 2009
- Stiller, C.; Lulcheva, I.:** Probabilistische Inferenz für Kognitive Automobile. In: Proceedings des 18. Workshop Computational Intelligence, Universitätsverlag Karlsruhe, 2008
- Sukthankar, R.:** Situation Awareness for Tactical Driving, Dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, 1997
- Tölle, W.:** Ein Fahrmanöverkonzept für einen maschinellen Kopiloten, VDI-Verlag, Düsseldorf, 1996
- Ulbrich, S.; Maurer, M.:** Probabilistic Online POMDP Decision Making for Lane Changes in Fully Automated Driving. In: Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems, 2013
- van Zanten, A.; Kost, F.:** Bremsenbasierte Assistenzfunktionen. In: Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015
- Viereckl, R.; Ahlemann, D.; Assmann, J.; Bratzel, S.:** Racing Ahead - The Connected C@r 2014 Study, PwC strategy&, 2014
- Vogel, O.; Mehlig, U.; Neumann, T.; Arnold, I.; Chughtai, A.; Völter, M.; Zdun, U.:** Software-Architektur, 2. Auflage, Spektrum Akademischer Verlag, Heidelberg, 2009
- Vollrath, M.; Briest, S.; Schießl, C.; Drewes, J.; Becker, U.:** Ableitung von Anforderungen an Fahrerassistenzsysteme aus Sicht der Verkehrssicherheit. In: BASt-Bericht F 60, Bundesanstalt für Straßenwesen, Bergisch-Gladbach, 2006
- Weilkiens, T.:** Systems Engineering mit SysML/UML, dpunkt-Verl, Heidelberg, 2008
- Weitzel, A.; Winner, H.; Peng, C.; Geyer, S.; Lotz, F.; Sefati, M.:** Absicherungsstrategien für Fahrerassistenzsysteme mit Umfeldwahrnehmung, BASt-Bericht F 98, Bundesanstalt für Straßenwesen, Bergisch Gladbach, 2014
- Werling, M.:** Ein neues Konzept für die Trajektoriengenerierung und -stabilisierung in zeitkritischen Verkehrsszenarien, Dissertation. In: Schriftenreihe des Instituts für Angewandte Informatik / Automatisierungstechnik des Karlsruher Instituts für Technologie, Band 34, 2010
- Wilhelm, U.; Ebel, S.; Weitzel, A.:** Funktionale Sicherheit und ISO 26262. In: Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015

-
- Wille, J. M.:** Manöverübergreifende autonome Fahrzeugführung in innerstädtischen Szenarien am Beispiel des Stadtpilotprojekts, Dissertation, Fakultät für Elektrotechnik, Informationstechnik, Physik der TU Braunschweig, 2012
- Wille, J. M.; Homeier, K.; Nothdurft, T.; Ohl, S.; Sasse, A.; Saust, F.; Hecker, P.; Maurer, M.; Schumacher, W.; Wolf, L.:** Der Stadtpilot : Autonomes Fahren auf dem Braunschweiger Stadtring, Institut für Regelungstechnik der TU Braunschweig, 2011
- Winner, H.:** Adaptive Cruise Control. In: Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015
- Winner, H.:** Grundlagen von Frontkollisionsschutzsystemen. In: Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015
- Winner, H.:** Quo vadis, FAS? In: Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015
- Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.:** Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015
- Winner, H.; Lotz, F.; Bauer, E.; Konigorski, U.; Schreier, M.; Adamy, J.; Pfromm, M.; Bruder, R.; Lüke, S.; Cieler, S.:** PRORETA 3: Comprehensive Driver Assistance by Safety Corridor and Cooperative Automation. In: Handbook of Driver Assistance Systems, Springer, Schweiz, 2016
- Ziegler, J.; Bender, P.; Lategahn, H.; Schreiber, M.; Strauß, T.; Dang, T.; Stiller, C.:** Kartengestütztes automatisiertes Fahren auf der Bertha-Benz-Route von Mannheim nach Pforzheim. In: Tagungsband des 9. Workshop Fahrerassistenzsysteme, Walting, 2014

Eigene Veröffentlichungen

Bauer, E.; Lotz, F.; Pfromm, M.; Schreier, M.; Cieler, S.; Eckert, A.; Hohm, A.; Lüke, S.; Rieth, P.; Abendroth, B.; Willert, V.; Adamy, J.; Bruder, R.; Konigorski, U.; Winner, H.: PRORETA 3: An Integrated Approach to Collision Avoidance and Vehicle Automation. at – Automatisierungstechnik 12/2012, Oldenbourg Wissenschaftsverlag, München, 2012

Hohm, A.; Lotz, F.; Fochler, O.; Lüke, S.; Winner, H.: Automated Driving in Real Traffic: from Current Technical Approaches towards Architectural Perspectives. In: SAE Technical Paper 2014-01-0159, SAE World Congress 2014, Detroit, USA, 2014

Karg, M.; Lotz, F.; Hirschle, A.; Hölig, J.: Manöverprädiktion als Baustein für sicheres und kooperatives Fahren. In: ATZ Automobiltechnische Zeitschrift, Ausgabe 07-08 2016, Springer Vieweg, Wiesbaden, 2016

Lotz, F.: System Architectures for Automated Vehicle Guidance Concepts. In: Maurer, M.; Winner, H.: Automotive Systems Engineering, Springer-Verlag, Berlin, 2013

Lotz, F.; Winner, H.: Maneuver Delegation and Planning for Automated Vehicles at Multi-Lane Road Intersections. In: Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems, Qingdao, China, Oktober 2014, 2014

Weitzel, A.; Winner, H.; Peng, C.; Geyer, S.; Lotz, F.; Sefati, M.: Absicherungsstrategien für Fahrerassistenzsysteme mit Umfeldwahrnehmung, BASt-Bericht F 98, Bundesanstalt für Straßenwesen, Bergisch Gladbach, 2014

Winner, H.; Lotz, F.; Bauer, E.; Konigorski, U.; Schreier, M.; Adamy, J.; Pfromm, M.; Bruder, R.; Lüke, S.; Cieler, S.: PRORETA 3 – An Integrated ADAS Concept – Comprehensive Driver Assistance by Safety Corridor and Cooperative Automation. In: 1. Internationale ATZ-Fachtagung Fahrerassistenzsysteme, Frankfurt a. M., Deutschland, April 2015, 2015

Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015

Winner, H.; Lotz, F.; Bauer, E.; Konigorski, U.; Schreier, M.; Adamy, J.; Pfromm, M.; Bruder, R.; Lüke, S.; Cieler, S.: PRORETA 3: Comprehensive Driver Assistance by Safety Corridor and Cooperative Automation. In: Handbook of Driver Assistance Systems, chapter 58, pages 1449–1469. Springer, Schweiz, 2016

Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbook of Driver Assistance Systems. Springer, Schweiz, 2016

Betreute studentische Arbeiten

Epenstein, Steffen: Entwicklung einer simulationsbasierten Testmethodik für automatisierte Fahrzeugführungskonzepte unter Berücksichtigung realer Teststrecken, Bachelor-Thesis Nr. 1110/12, 2012

Graupner, Maren: Bewertung des Standes der Forschung zum Kooperationsverhalten im Straßenverkehr und Ableitung von Anforderungen an das Kooperationsverhalten bei Nutzung höher automatisierter Fahrzeuge, Master-Thesis Nr. 546/13, 2013

Münker, Tobias: Entwicklung und Validierung einer Methodik zur messdatenbasierten Parametrisierung eines Fahrzeugmodells zur Fahrdynamiksimulation mit IPG CarMaker, Master Thesis Nr. 499/12, 2012

Pfeifer, Roman: Ermittlung einer Methodik zur systematischen Generierung von Anwendungsszenarien für ein Fahrerassistenz- und Fahrsicherheitssystem, Studienarbeit Nr. 1085/11, 2011

Pfeifer, Roman: Entwicklung und Verifikation eines situativ gesteuerten Verhaltensplaner-Moduls für die teilautomatisierte und kooperative Fahrzeugführung, Diplomarbeit, Institut für Arbeitswissenschaft, 2013

Puphal, Tim: Entwicklung und Implementierung einer Methodik zur Generierung von Testfällen für die automatisierte Fahrzeugführung durch Identifikation der Systemgrenzen, Bachelor-Thesis Nr. 1151/13, 2013

Schönichen, Jonas: Integration von Informationen digitaler Karten in ein Fusions-Framework zur Verbesserung der Umfeldwahrnehmung für Fahrerassistenz-Konzepte. Master-Thesis, Fachgebiet Regelungstheorie und Robotik der TU Darmstadt, 2012

Lebenslauf

Persönliche Daten

Name: Felix Gerhard Otto Lotz
Geburtsdatum: 04.11.1983
Geburtsort: Alzenau

Schulzeit

1994 – 2003 Spessart-Gymnasium, Alzenau

Studium

2004 – 2010 Studium Allgemeiner Maschinenbau
Technische Universität Darmstadt
08/2008 – 12/2008 Auslandssemester, Virginia Polytechnic Institute and
State University, Blacksburg, VA, USA
08/2010 Abschluss als Dipl.-Ing.

Berufstätigkeit

01/2011 – 12/2014 Wissenschaftlicher Mitarbeiter am Fachgebiet Fahrzeug-
technik der Technischen Universität Darmstadt; Bearbei-
ten des Forschungsprojektes PRORETA 3 in Kooperation
mit der Continental AG
seit 03/2015 Projektleiter Funktionsentwicklung automatisiertes Fah-
ren bei der Continental AG, Frankfurt a. M.